

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Tomaž Biber

**Vrednotenje socio-tehničnih vidikov  
elementov metodologije razvoja  
programske opreme**

DIPLOMSKO DELO  
NA INTERDISCIPLINARNEM UNIVERZITETNEM ŠTUDIJU

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana, 2016



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko in mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Vrednotenje socio-tehničnih vidikov elementov metodologije razvoja programske opreme

Evaluation of social-technical aspects of software development methodologies

Tematika naloge:

V okviru diplomske naloge najprej pripravite krajši pregled področja metodologij razvoja programske opreme, pri čemer se osredotočite na njihovo sestavo in življenjske modele. Predstavite tudi prednosti in slabosti različnih pristopov, pri čemer upoštevajte tako tehnične kot sociološke vidike. Nato kratko predstavite model za vrednotenje socio-tehničnih vidikov metodologij ter ga z ustreznimi prilagoditvami uporabite za ovrednotenje metodologije razvoja programske opreme v izbranem podjetju. Tako pridobljene rezultate komentirajte in ovrednotite. Kritično ovrednotite tudi koristnost evalvacije za potrebe izboljševanja procesa razvoja programske opreme v izbranem podjetju.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Tomaž Biber sem avtor diplomskega dela z naslovom:

*Vrednotenje socio-tehničnih vidikov elementov metodologije razvoja programske opreme.* S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 1.julij 2016

Podpis avtorja:





*Zahvaljujem se mentorju doc. dr. Damjanu Vavpotiču za komentarje in  
napotke pri pisanju diplomske naloge.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Metodologije razvoja programske opreme</b>	<b>3</b>
2.1	Neuspešni razvojni projekti . . . . .	4
2.2	Metamodel metodologije razvoja IS . . . . .	6
2.2.1	Ljudje . . . . .	8
2.2.2	Proces . . . . .	10
2.3	Lahke in težke metodologije . . . . .	11
2.4	Spredaj in zadaj utežene metodologije . . . . .	12
<b>3</b>	<b>Proces in modeli razvoja IS</b>	<b>15</b>
3.1	Zaporedni ali slapovni model . . . . .	16
3.2	Iterativni model . . . . .	18
3.3	Inkrementalni model . . . . .	19
3.4	Agilne metodologije . . . . .	21
<b>4</b>	<b>Model vrednotenja socio-tehničnih vidikov metodologije razvoja programske opreme</b>	<b>25</b>
4.1	Ogrodje vrednotenja metodologije razvoja . . . . .	26
4.2	Lastnosti sociološke sprejetosti . . . . .	30
4.3	Lastnosti tehnične učinkovitosti . . . . .	31

<b>5</b>	<b>Vrednotenje metodologije znotraj opazovane razvojne skupine</b>	<b>35</b>
5.1	Vloge in naziv članov razvojne skupine . . . . .	36
5.1.1	Mlajši razvijalec programske opreme . . . . .	36
5.1.2	Razvijalec programske opreme . . . . .	37
5.1.3	Izkušeni razvijalec programske opreme . . . . .	39
5.1.4	Arhitekt poslovnih rešitev . . . . .	41
5.2	Procesni diagram poteka . . . . .	43
5.3	Merilni inštrument . . . . .	45
<b>6</b>	<b>Analiza rezultatov</b>	<b>47</b>
6.1	Prikaz rezultatov vrednotenja socio-tehničnih dimenzij za vse elemente . . . . .	48
6.2	Prikaz ovrednotenja posameznega EMR . . . . .	51
6.2.1	Razvoj SSRS poročil (Q9 ) . . . . .	51
6.2.2	Osnutek za oddajo ponudb (Q4 ) . . . . .	53
6.2.3	Performančna analiza (Q22 ) . . . . .	54
<b>7</b>	<b>Sklepne ugotovitve</b>	<b>57</b>
	<b>Literatura</b>	<b>59</b>

# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>IS</b>	information system	informacijski sistem
<b>EMR</b>	development methodology element	element metodologije razvoja
<b>ERP</b>	enterprise resource planning	programske opreme za upravljanje poslovnih virov
<b>DoD</b>	definition of done	definicija končane zahteve
<b>DoR</b>	definition of ready	definicija pripravljene zahteve
<b>FRD</b>	functional requirements document	dokument specifikacije zahtev
<b>AOS</b>	application object server	aplikacijski strežnik
<b>AOT</b>	application object tree	drevo aplikacijskih objektov



# Povzetek

**Naslov:** Vrednotenje socio-tehničnih vidikov elementov metodologije razvoja programske opreme

V diplomskem delu smo predstavili različne metodologije razvoja programske opreme in model za vrednotenje tehničnih in socioloških vidikov metodologije razvoja programske opreme. Model smo uporabili v praksi v podjetju znotraj razvojne skupine na oddelku za implementacijo ERP-rešitve Microsoft Dynamics AX. Analizirali smo obstoječo metodologijo in med člani razvojne skupine uporabili inštrument, na podlagi katerega smo ovrednotili posamezne elemente metodologije razvoja in jih postavili v ogrodje modela za vrednotenje metodologije. Kot rezultat uporabe modela za vrednotenje metodologije smo izpostavili neučinkovite, nesprejete in neuporabne elemente.

**Ključne besede:** Metodologija, metamodel metodologije, proces, življenjski cikel, zaporedni model, iterativni model, inkrementalni model, agilne metodologije, socio-tehnična evalvacija.





# Abstract

**Title:** Evaluation of social-technical aspects of software development methodologies

The thesis presents different software development methodologies (SDM) and an evaluation model for SDM suitability on a technical and social dimension. Model was used in the company within the development team at the department for implementation of ERP solution Microsoft Dynamics AX. We have analyzed the existing methodology and between members of the team we used the instrument for EMR evaluation and placed EMR in the SDM framework model for evaluation methodology . As a result of the use of evaluation model we identified useless, inefficient and unadopted SDM elements.

**Keywords:** Methodology, metamodel of methodology, process, software development life cycle, waterfall model, iterative model, incremental model, agile software development, social-technical evaluation .



# Poglavje 1

## Uvod

Metodologije razvoja informacijskih sistemov so se skozi desetletja precej spreminjale in se spreminjajo še danes. Za enega izmed prvih formalnih opisov metodologije se pogosto navaja leto 1970, ko je Winston W. Royce predstavil temelje za zaporedni model. Metodološke smernice so se v tem obdobju zelo spreminjale. V diplomskem delu bomo zato formalizirali pojem metodologije in predstavili nekaj načinov za delitev metodologij glede na njihove lastnosti. Predstavili bomo pomen ljudi in procesa znotraj metamodela metodologije razvoja IS.

V naslednjem poglavju bomo opisali štiri najpogostejše modele razvoja IS. Zavedanje značilnosti, prednosti in slabosti posameznega modela razvoja IS nam bo v pomoč v 5. poglavju pri vrednotenju metodologije.

Praktični del diplomskega dela smo izvedli v večjem slovenskem računalniškem podjetju, kjer smo v razvojni skupini znotraj oddelka za implementacijo sistema za upravljanje poslovnih rešitev (ERP) Microsoft Dynamics AX naredili analizo. Microsoft Dynamics AX je sistem za upravljanje poslovanja (ERP), ki pokriva proizvodnjo, maloprodajo, distribucijo, upravljanje financ, upravljanje s človeškimi viri in področje javnega sektorja. Vgrajene vertikalne rešitve za različne industrije pripomorejo k visoki učinkovitosti produkta v različnih poslovnih okoljih. Opazovana razvojna skupina izvaja razvoj novih in nadgradnjo obstoječih funkcionalnosti znotraj produkta Mi-

crosoft Dynamics AX.

V četrtem poglavju bomo predstavili model za vrednotenje tehničnih in socioloških vidikov metodologije. Model se uporablja za sistematično ovrednotenje posameznih sestavnih elementov metodologije (EMR). Z uporabo modela pridobimo informacije o sociološki sprejetosti in tehnični ustreznosti EMR. Model definira množice lastnosti in množice razlogov za ocenjene vrednosti lastnosti za opazovani EMR iz sociološkega in tehničnega vidika.

Cilj diplomskega dela je uporaba modela za vrednotenje metodologije v opazovani razvojni skupini in pridobiti nove informacije, ki so lahko podlaga za nadaljnje izboljšave obstoječe metodologije. V petem poglavju smo zato najprej analizirali obstoječo metodologijo in zajeli ključne EMR. Uporabili smo model za vrednotenje metodologije in predstavili dobljene informacije o posameznih gradnikih metodologije. Izpostavili smo EMR, ki so bili ovrednoteni za tehnično nesprejemljive ali pa sociološko neustrezne.

V zadnjem poglavju (Zaključek) bomo podali ugotovitve, ki se navezujejo na uporabo modela za vrednotenje metodologije ter na oceno obstoječe metodologije v opazovanem podjetju.

## Poglavje 2

# Metodologije razvoja programske opreme

Cilj razvoja IS je ustvariti programsko opremo, katera ustreza potrebam naročnika in je zanesljiva, učinkovita, kakovostna ter trajnostna. Veliki kompleksni projekti, med katere uvrščamo tudi razvoj ERP-sistemov, so podvrženi velikemu tveganju neuspeha. Projekti morajo biti realizirani znotraj časovno predvidenih terminov ter znotraj predvidenega proračuna. Pogosto se zahteve naročnika med razvojem spreminjajo, oziroma nadgrajujejo. Ker zadostiti vsem zahtevam ni lahka naloga, se je pojavila potreba po bolj strukturiranem pristopu razvoja IS, kjer bodo dobro definirani postopki, predpisane metode dela, tehnike in orodja.

Definicijo metodologije razvoja IS lahko podamo v obliki naslednjih trditev.

**Trditev 2.0.1** *Metodologija je priporočena zbirka filozofij, faz, postopkov, pravil, tehnik, orodij, dokumentacije, upravljanja in izobraževanja za razvijalce IS [6].*

**Trditev 2.0.2** *Metodologija razvoja IS je priporočen način razvoja IS, ki temelji na filozofiji in množici principov [1].*

Iz same definicije metodologije je razvidno, da ima pomembno mesto

znotraj metodologije sociološka komponenta. Prav ljudje so v metodologiji ključnega pomena, kajti metodologija je namenjena njim, zato ni možno, da bi nastala neodvisno od ljudi. Pridobljena znanja, izkušnje, osvojeni principi dela so osnova za formalizacijo metodologije. Razvoj ljudi znotraj projektne skupine bogati samo metodologijo, saj vpliva na njeno formalizacijo in samo uporabo metodologije.

Metodologija niso tako samo predpisani procesi, orodja, temveč opredeljuje tudi podporne postopke, pravila odločanja, načine komunikacije znotraj razvojne skupine. Zato lahko metodologijo definiramo tudi kot množico dogovorov, s katerimi se projektna skupina strinja [4]. Vsaka metodologija ni primerna za vsako vrsto projekta, zato je pomembno, da si jo organizacije prilagodijo glede na njihov način dela ter področje dela.

## 2.1 Neuspešni razvojni projekti

IS lahko organizacijam prinašajo pomembno konkurenčno prednost. Dotikajo se različnih področij znotraj organizacije in postajajo vedno večji. Izpad IS ali napake znotraj IS predstavljajo veliko tveganje za celotno organizacijo.

Neuspešen projekt je projekt, kateri ne dosega pričakovanih rezultatov. Tak projekt lahko vsebuje naslednje lastnosti:

- Prekoračitev proračuna
- Prekoračitev časovnega okvira
- Ne ustreza potrebam naročnika
- Prezahteven za uporabo
- Kvaliteta slabša od predvidene
- Slabša učinkovitost od predvidene

Dovolj je, da je prisotna samo ena izmed naštetih lastnosti in že lahko povzroči, da projekt označimo za neuspešnega. Za tak primer bi lahko vzeli



**Slika 2.1:** % neuspešnih projektov razvoja IS

projekt, v okviru katerega je bil razvit kakovosten IS, ki ustreza potrebam naročnika, vendar pa je prekoračil proračun in s tem ogrozil obstanek podjetja.

Mnogo razvojnih projektov se nikoli ne konča. Tudi končani projekti so lahko neuspešni. Pogosta pojava sta prekoračitev proračuna in prekoračitev časovnih okvirov projekta. V raziskavo [10], ki jo je leta 2012 pripravilo podjetje McKinsey v sodelovanju z univerzo v Oxfordu, je bilo vključenih 5400 velikih informacijskih projektov, katerih proračun je presegal 15 milijonov \$. Ugotovljeno je bilo, da je kar 45 % projektov prekoračilo finančni proračun. Več kot 7 % projektov je prekoračilo dogovorjeni časovni okvir in dostavilo manj kot 56 % pričakovane vrednosti. Daljši kot je časovni okvir projekta, večja je verjetnost, da bo projekt neuspešen. Približno za 15 % se poveča možnost za neuspeh in prekoračitev proračuna na dodatnem planiranem letu razvoja IS. Projekti, ki prekoračijo pri razvoju IS proračun za več kot 200 %, so lahko usodni za obstoj podjetja. Ocenjeno je, da je tako močno neuspešnih projektov približno 17 %.

Razlogi, ki najpogosteje vplivajo na slab končni rezultat in neuspešen projekt so [5]:

- Slabo vodenje projekta
- Nerealno postavljeni cilji
- Slabo definirane zahteve IS
- Netočna ocena potreb po človeških virih za izvedbo projekta
- Neupravljanje s tveganji

- Slabo poročanje vmesnih statusov
- Slaba komunikacija med stranko in projektno skupino
- Nezmožnost upravljanja projektne kompleksnosti
- Neučinkovito razvojno okolje
- Neučinkovita razvojna skupina
- Nedorečeni postopki testiranja
- Slaba metodologija razvoja IS

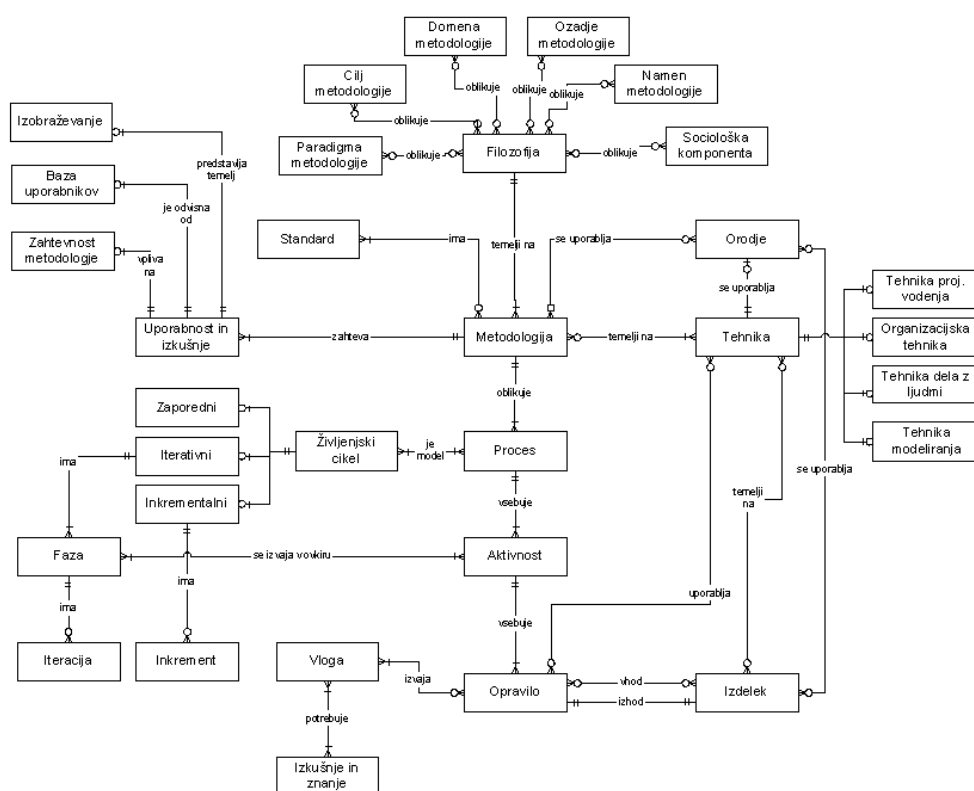
Znotraj neuspešnih projektov je tako zaznati prisotnost različnih razlogov, ki se med seboj prepletajo. Klub vsemu pa se večji del projektov konča uspešno. Potrebno se je zavedati omejitev in pasti, katere prinaša sam razvoj IS. Ob poznavanju lastnosti neuspešnih projektov ter razlogov, ki privedejo do negativnega končnega rezultata, lažje razumemo pomen dobre razvojne metodologije.

## 2.2 Metamodel metodologije razvoja IS

Razvoj IS je lahko zelo kompleksen in predstavlja močan izziv. V prejšnjem poglavju smo izpostavili lastnosti neuspešnih projektov in s tem predstavili nekaj rizičnih dejavnikov. Posamezne elemente metodologije bomo predstavili v obliki metamodela [8].

Metamodel predstavljajo elementi v obliki entitetnih tipov in množica relacij med njimi. Glavni element je metodologija, ki temelji na filozofiji in tehnikah, zahteva uporabnost in izkušnje, oblikuje jo proces, uporablja orodja in ima predpisane standarde. Elementi, ki oblikujejo filozofijo, na kateri temelji metodologija, so: cilji metodologije, domena metodologije, ozadje metodologije, namen metodologije, sociološka komponenta in paradigma metodologije. Na uporabnost metodologije vpliva sama stopnja zahtevnosti





Slika 2.2: Metamodel metodologije razvoja IS. Vir [8].

metodologije in je odvisna od izkušenj in znanja baze uporabnikov. Metodologijo oblikuje proces, s katerim je definiran življenjski cikel razvoja programske opreme. Življenjski cikel je lahko zaporedni, iterativni ali inkrementalni. Na podlagi samega načina izvajanja življenjskega cikla in procesa razvoja lahko govorimo o različnih modelih metodologije. Glede na model se posamezne faze življenjskega cikla izvajajo inkrementalno ali iterativno. Aktivnosti se izvajajo znotraj posameznih faz in predstavljajo del osnovnega procesa. Aktivnosti vsebujejo množice opravil, ki jih izvajajo ljudje s predpisanimi vlogami in potrebnimi izkušnjami ter znanjem. Opravilo lahko uporablja tehnike in kot rezultat opravljenega dela izstavi izdelek.

Vsi elementi metodologije nimajo enake teže v vsakem modelu metodologije. Potrebno je poiskati pravo ravnotežje med elementi. Obstajajo tudi nekoliko drugače obravnavane metodologije, ki jih sestavljajo tudi drugi elementi, ki niso vključeni v sliki 2.2. Nekatere od teh metodologij dajejo velik poudarek ljudem. Ker v opazovani razvojni skupini uporabljajo prilagojeno agilno metodologijo, smo se odločili, da bomo v diplomskem delu izpostavili dva elementa: ljudi in proces [5].

### **2.2.1 Ljudje**

Ljudje so najpomembnejši vir v razvojnem projektu. Njihov vložen trud in znanje bodo na koncu projekta dostavila delujočo programsko opremo. Organizacije strmijo k usposobljenim, izobraženim in motiviranim zaposlenim. Ljudem je v projektnem delu potrebno zagotoviti možnost dodatnega izobraževanja ter jim pomagati pri njihovi osebni rasti. Ljudi bi lahko označili kot osrčje vsakega uspešnega projekta. Le dobro organizirana skupina inženirjev, ki imajo znanja in izkušnje, je sposobna uspešno opraviti vse aktivnosti na projektu. Dobre alternative ni in pogosto vodi v kaos in neuspešen projekt.

Ljudje so na projektih organizirani v različne skupine. Vsaka skupina pokriva svoje področje in je vključena v različne procese. Številne vloge so zato vezane na posamezno skupino. Za pravilno izvajanje opravil je pred-

pisan nabor znanja in uporabniških izkušenj. Kljub temu da skupine izvajajo različne aktivnosti v procesu, imajo enak skupni končni cilj po uspešnem izdelku. Ljudi na projektu bi lahko razdelili v pet skupin [5]:

- Vodstvo

Ljudje so vključeni v proces vodenja podjetja, ki se ukvarja z razvojem programskih rešitev. Njihova glavna funkcija je zagotavljati dobiček, stroškovno učinkovitost, konkurenčnost na trgu in zadovoljstvo strank. Vsebinsko niso vključeni v projekt, prav tako tudi tehnični vidik projekta ni njihovo področje.

- Vodje projektov

Vodje projektov so odgovorne za planiranje in sledenje projektu. Njihova glavna naloga je upravljati z ljudmi, aktivnostmi in procesi. V času izvajanja projekta po potrebi uvajajo novosti in izboljšave ter skrivajo, da projekt ostane znotraj predvidenega proračuna in časovnega okvira.

- Razvojna skupina

Razvojni inženirji so odgovorni za razvoj programske opreme. Njihova naloga ni zgolj kodiranje novih funkcionalnosti, temveč so vključeni v številne aktivnosti, kot so: načrtovanje tehničnega oblikovanja rešitve, pisanje dokumentacije, proces testiranja, vzpostavitev infrastrukture in konfiguracija programske opreme. Pomembno je, da so člani razvojne skupine med seboj usklajeni in visoko motivirani. Razvojno skupino lahko sestavljajo ljudje z različnim nivojem znanja in izkušenj, zato lahko predstavlja velik izziv, kako ljudi med seboj povezati v usklajeno celoto. Dobro delovno okolje, vključenost novih tehnologij v projekte, imeti čas in podporo pri izobraževanju in spremljanju informacijskih novosti, nagrajevanje in napredovanje, so prav gotovo pomembni dejavniki, ki vplivajo na motivacijo razvojnega inženirja.

- Stranka

Stranka je odgovorna za nakup programske opreme. Vključena je v fazo zajema zahtev. Dobro definirani poslovni procesi novega IS in natančno definirane strankine zahteve imajo velik vpliv na uspešnost projekta. Interes stranke je dobra programska oprema, ki je stroškovno učinkovita, zadostuje poslovnim potrebam in je visoke kakovosti.

- Končni uporabniki

Končni uporabniki so ljudje, ki uporabljajo programsko opremo. Želijo si uporabljati preprost sistem ob enem pa visoko učinkovit. Delovni procesi končnih uporabnikov bi se morali z uvedbo novega IS optimizirati in končnim uporabnikom olajšati vsakodnevno delo.

## 2.2.2 Proces

Proces je okvir, ki skrbi, da se aktivnosti na projektu izvajajo organizirano in na discipliniran način. Pomaga usmerjati ljudi pri njihovih opravilih. Sam proces razvoja med izvajanjem projekta prehaja med različnimi fazami, ki so med seboj povezane in časovno omejene. Proces oblikuje povezave med posameznimi fazami in definira medsebojne povezave med fazami in njihovo pogostostjo. Specifični proces IS imenujemo tudi življenjski cikel, za katerega uporabljamo tudi izraz model. Obstaja več različnih modelov, večina pa temelji na zaporednem ali iterativnem modelu [13].

Zaporedni ali spalovni model je najpreprostejši model in predstavlja osnovo za večino ostalih modelov. Prehajanje med fazami poteka zaporedno. Ko se ena faza v celoti konča, se začne naslednja. Faze se izvajajo v točno določenem vrstnem redu. Ko se posamezna faza konča, se predpostavi, da so aktivnosti, vezane na fazo, končane in ni potrebe po ponovnem izvajanju te faze. Slapovni model se začne s fazo analize in kot produkt faze nastane dokument specifikacija zahtev. Sledi faza načrtovanja, ki sproducira dokument načrt. Naslednja faza je izvedba, ki dostavi kodo programske rešitve.

Fazi implementacije sledi testiranje ter nato faza vzdrževanja. Življenjski cikel se pri zaporednem modelu izvede le enkrat.

Razvoj IS redko poteka strogo po zaporednem modelu. V praksi se pogosto uporablja iterativni model, kjer se posamezne faze zaporednega modela večkrat ponovijo. Nekateri procesi se lahko izvajajo tudi vzporedno. Pri agilnih modelih so prehodi med fazami lahko bolj zabrisani in se faze ponavljajo v iteracijah, na primer na dva tedna. Posamezne modele, njihove lastnosti, prednosti in slabosti bomo opisali v poglavju 3.

## 2.3 Lahke in težke metodologije

S pojavom novih oblik razvojnih metodologij je nastala delitev metodologije glede na težo. Z izrazom težke in lahke metodologije opisujemo ravnovesje med obsegom in gostoto metodologije. Večji kot je obseg metodologije, več je vključenih procesov, boljše so definirane tehnike in orodja, več standardov je vključenih. Z gostoto označimo raven predpisane formalnosti posameznih EMR in nivo podrobnosti. Metodologije z visoko gostoto imajo podrobno opisane procese, predpisana zahtevana znanja in izkušnje za izvajanje opravil, podrobno opisujejo izdelavo dokumentacije in niso prilagodljive uporabniku.

Lahke metodologije so metodologije z manjšo gostoto in manjšim obsegom. Takšne metodologije so uporabne pri majhnih projektih, kjer je glavni cilj delujoča programska oprema. Pri velikih projektih, kjer je rizičnih elementov za neuspeh projekta več, kjer so razvojne skupine velike, ta metodologija ni optimalna. Lahke metodologije so primerne v razvojnih skupinah, kjer imamo izkušene in motivirane inženirje. Velik pozitiven učinek imajo lahke metodologije v projektih, kjer so procesi nepredvidljivi in se zahteve strank med razvojem spreminjajo. Visoka stopnja prilagodljivosti zahteva izkušeno in motivirano razvojno skupino, katera obvladuje področje dela. Če je metodologija visoko prilagodljiva, sledi kot posledica nizka optimiziranost. Sam proces izdelave dokumentacije ni podrobno definiran in kot cilj opredeljuje le razvoj programske rešitve. Stranka, ki je vključena v tak projekt, mora

razumeti filozofijo metodologije in skladno z njo tudi sodelovati.

Težke metodologije so metodologije z večjo gostoto in večjim obsegom. Takšne metodologije so visoko optimizirane in nizko prilagodljive. Primerne so za velike projekte, kjer so zahteve strank relativno stabilne in se med razvojem bistveno ne spreminjajo. Primer takega projekta je uvedba novega ERP v podjetju, kjer se poleg nove programske rešitve optimizirajo tudi procesi v organizaciji. Za razvoj velikih projektov so potrebne večje razvojne skupine in visoka stopnja formalizma opredeljuje opravila in vloge, vezane na posamezne člane skupine. Težke metodologije predpisujejo postopke za izdelavo dokumentacij, predloge dokumentov in uporabo predpisanih tehnik in orodij. Dobra tehnična in uporabniška dokumentacija je v fazi uvedbe in kasneje v fazi vzdrževanja zelo pomembna.

## 2.4 Spredej in zadaj utežene metodologije

Metodologije razvoja IS določajo, kakšen poudarek bo namenila razvojna skupina posamezni fazi življenjskega cikla. Če metodologija predpisuje večjo formalizacijo za fazi analize in načrtovanja, lahko govorimo o spredej uteženi metodologiji. Metodologija, ki se osredotoča na fazo razvoja in testiranja, pa je zadaj utežena metodologija. Utežitev metodologije definira, katere aktivnosti v procesu razvoja IS so poudarjene [2].

Spredej utežene metodologije podrobno opredeljujejo specifikacije uporabniških zahtev, zahtev po IT infrastrukturi, natančen plan dela in ostale dokumente, povezane z analizo in načrtovanjem. Ker sta prvi dve fazi razvojnega cikla tako podrobno predpisani, se pričakuje, da bo sama implementacija potekala tehnično dovršeno in kot rezultat podala IS z malo napakami. Ponavadi se uvrščajo spredej utežene metodologije med težke metodologije. Takšen pristop je primeren za razvoj kritičnih sistemov, kjer želimo minimizirati možnost napake v samem izdelku. Vendar pa je potrebno imeti v naprej dobre definirane uporabniške zahteve, saj takšna metodologija ni prilagojena za projekte, kjer se zahteve spreminjajo. Včasih pa se zahteve podrob-

neje opredelijo v kasnejši fazi implementacije. V takšnih primerih spredaj utežena metodologija ni optimalna, saj s kasnejšo uvedbo sprememb lahko poruši celotno zasnovano arhitekturno postavitev programske opreme. Velike razvojne skupine lahko zaznajo prednost v spredaj uteženih metodologijah, saj si lažje razdelijo delo med člani skupine v fazi implementacije.

Zadaj utežene metodologije postavljajo pomen kodiranja, fazo izvedbe in testiranja pred analizo in načrtovanjem. Takšen pristop se je v zadnjem obdobju zelo uveljavil v majhnih razvojnih skupinah pri razvoju nekritičnih projektov. V proces izvedbe so vključene nekatere aktivnosti iz faze analize in načrtovanja, vendar brez dodatne dokumentacije. V projektih, kjer so zahteve slabo podane, so zadaj utežene metodologije uspešne. Porabljen čas za spremembe tako ne zraste eksponentno kot pri klasičnem modelu. Slabost takšnega pristopa pa je lahko nestabilna programska oprema, katero je težko vzdrževati in nadgrajevati. Inženirji, ki so vključeni v proces, morajo biti visoko motivirani in izkušeni, saj njihov plan dela ni strogo definiran. Zadaj utežene metodologije so primerne tudi za projekte, kateri uporabljajo hitro spreminjajoče se tehnologije.





## Poglavje 3

# Proces in modeli razvoja IS

Proces razvoja IS sestavljajo različne aktivnosti, ki se začnejo s konceptualno zasnovo rešitve in nadaljujejo vse do izstavitve programske opreme. Običajno se projekte organizira v razvojne faze, znotraj katerih se izvajajo predpisane aktivnosti. Proces definira povezave med razvojnimi fazami, vrstni red izvajanja, njihovo ponovljivost, predpisuje pogoje za prehajanja med fazami in izhodne izdelke faz, ki so del metodologije [12]. Poleg aktivnosti, katere predpisuje proces za posamezno fazo razvoja IS, obstajajo še krovne aktivnosti. Krovne aktivnosti se neprenehoma izvajajo skozi celoten cikel razvoja in med njih uvrščamo aktivnosti projektnega vodenja, upravljanja rizika in upravljanje kakovosti. Za specifično definiran proces IS uporabljamo tudi izraz model. Večina modelov predpisuje podobne množice aktivnosti in faz. Glavne faze razvoja IS so:

- Analiza
- Načrtovanje
- Implementacija
- Testiranje
- Uvedba
- Vzdrževanje

Model razvoja IS definira vrstni red in pogostost izvajanja faz v razvojnem projektu. Posamezni modeli so se izoblikovali med razvojem informacijskih tehnologij in temeljijo na različnih filozofijah. Vsak model razvoja IS ni ustrezen za vse projekte, zato bomo predstavili zaporedni ali slapovni model, iterativni model, inkrementalni model in agilni model. Zavedanje značilnosti, prednosti in slabosti posamezne metodologije nam bo v pomoč v 5. poglavju pri vrednotenju metodologije znotraj oddelka za razvoj ERP v slovenskem podjetju [3].

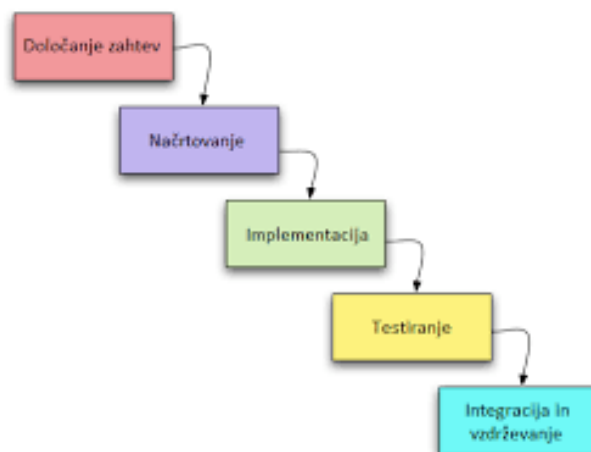
### 3.1 Zaporedni ali slapovni model

Leta 1970 je Royce definiral enega izmed najstarejših modelov, imenovanega zaporedni model. Zaporedni model imenujemo tudi slapovni, saj pristop razvoja vidimo kot slap, ki teče navzdol skozi več faz. Kljub nekaterim pomanjkljivostim je danes zaporedni model še vedno široko v uporabi in je temelj za mnoge druge modele. Na sliki 3.1 so prikazane posamezne faze zaporednega modela. Slapovni model predpisuje zaporedno izvajanje faz. Izhodni izdelek ene faze se uporablja kot vhodni izdelek druge faze, zato se naslednja faza lahko začne šele, ko se predhodna faza v celoti zaključi. Zaporedni model ne predvideva sočasnega izvajanja aktivnosti različnih faz. Ker si faze sledijo zaporedno in vračanje nazaj ni mogoče, je takšen model primeren za projekte, kjer so zahteve dobro specificirane in se med projektom bistveno ne spreminjajo. Zaporedni model omogoča projektni vodji dober pregled nad izvajanjem projekta.

Kot veliko omejitev slapovnega modela lahko izpostavimo izvajanje faze testiranja šele pri koncu razvojnega cikla. Ko prvič testiramo programsko opremo, jo testiramo že kot celoto.

Prednosti zaporednega modela:

- V uporabi je mnogo let, zato je med ljudmi široko sprejet. Ker je proces dobro definiran in razumljiv, ga ljudje sprejemajo.



**Slika 3.1:** Zaporedni ali slapovni model

- Je preprost in enostaven za uporabo. Kriteriji za prehode med fazami so dobro definirani, vrstni red faz je uporabnikom razumljiv.
- Olajša načrtovanje človeških virov po posamezni fazi, saj so potrebna znanja in izkušnje, potrebne za izvedbo aktivnosti, dobro opredeljene.
- Deluje dobro pri manjših projektih, kjer so zahteve dobro definirane in nespreminjajoče.
- Omogoča dobro projektno vodenje pod pogojem, da potek razvoja napreduje po načrtu.

Slabosti zaporednega modela:

- Ne predvideva paralelnega izvajanja aktivnosti v različnih fazah. Vsaka faza se mora v celoti zaključiti, preden nastopi nova faza.
- Vse zahteve morajo biti znane že ob začetku izvajanja projekta. Model ne predvideva dodatnih podrobnejših zahtev med izvajanjem načrtovanja in implementacije.
- Povratna informacija o programski opremi je predvidena šele v fazi

testiranja, zato se lahko veliki problemi pokažejo prepozno v razvojnem ciklu.

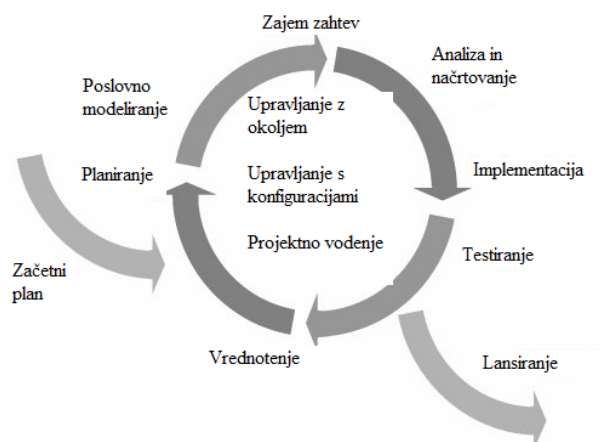
- Raba človeških virov je neučinkovita. Uporabniki so s svojimi vlogami lahko vključeni samo v določene faze in čakajo na zaključek predhodnih faz.

## 3.2 Iterativni model

Iterativni model je nastal kot posledica pomanjkljivosti zaporednega modela. Iterativni model ostaja na temeljih zaporednega modela, saj ohranja enake razvojne faze. Iterativni model rešuje problem zaporednega modela, kjer je potrebno v celoti izvesti analizo zahtev, v celoti kreirati načrt rešitve in narediti dokončno implementacijo znotraj ene faze. V fazi implementacije se pogosto pojavi potreba po izboljšavah, zato je pomembno, da obstaja možnost prahajanja med fazami tudi nazaj. Iterativni življenjski cikel je bolj naraven pristop k razvoju programske opreme, saj se posamezna faza razvoja izvaja v več iteracijah. V iteracijo so vključene poljubne faze razvoja, ponavlja pa se med izvajanjem ene iteracije izvedejo vse razvojne faze. Izkušnje in znanja, pridobljena iz predhodnih iteracij, povečujejo učinkovitost razvojnega cikla. Prav ta lastnost iterativnega modela pomaga pri upravljanju s tveganji ter objektivnem ocenjevanju projekta. Posamezne iteracije po končanem ciklu iteracije izstavijo del funkcionalnosti celotnega IS. V praksi se pogosto pojavljata iterativni in inkrementalni model skupaj. Sam iterativni model vedno ne zagotavlja učinkovitega pristopa, saj se lahko iteracije ponavljajo v nedogled, ne da bi pri tem nastajala celostna rešitev IS.

Prednosti iterativnega modela:

- Razvoj IS poteka po iteracijah, kar omogoča zgodnje odkrivanje odklonov od pričakovane rešitve.
- Omogoča merjenje napredka razvoja IS z ocenjevanjem izvedenega dela po končani iteraciji.



Slika 3.2: Iterativni model

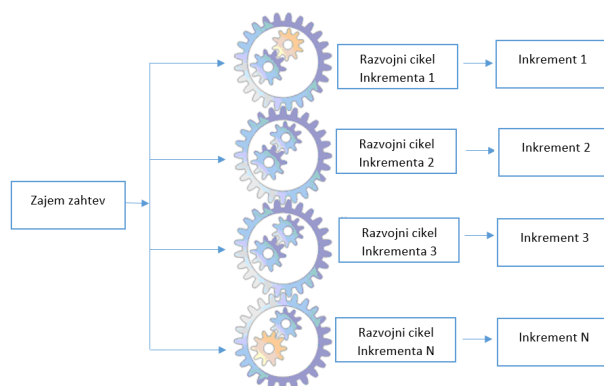
- Po končani iteraciji je možno pridobiti povratno informacijo s strani uporabnikov, ki lahko vpliva na nadaljnje iteracije.
- Razvoj ključnih, kritičnih funkcionalnosti IS se izvede v začetnih iteracijah razvoja.

Slabosti iterativnega modela:

- Vsaka faza iteracije je toga in brez prekrivanja.
- Povečana možnost nastanka dragih in velikih arhitekturnih sprememb, saj vse zahteve niso bile znane v naprej.
- Zahtevno vodenje projekta.
- Nesposobnost oceniti število potrebnih iteracij do končne rešitve IS.

### 3.3 Inkrementalni model

Inkrementalni model razdeli uporabniške zahteve na več sklopov in v postopku izvedbe IS sprotno predaja posamezni inkrement uporabniku. Razvojni cikli so razdeljeni na manjše, boljje obvladljive enote oziroma module. Razvoj



**Slika 3.3:** Inkrementalni model

vsakega modula gre skozi vse faze življenjskega cikla, to so: analiza, načrtovanje, implementacija in testiranje. Delovna verzija IS je dostopna že po prvem zaključenem modulu. Vsaka nova verzija IS objavi dodatni modul k prejšnjim in tako gradi celostni IS. Modul, katerega dobimo z uspešno izvedbo enega inkrementa, je potrebno integrirati s preostalimi že obstoječimi moduli delovne verzije IS. Posamezni moduli predstavljajo določeno funkcionalnost razvijajočega se sistema. V sistemu ERP so glavni moduli, ki jih pogosto vpeljujejo inkrementalno: nabava, prodaja, glavna knjiga, skladišče, proizvodnja in človeški viri. Vrstni red izvedbe posameznega modula je potrebno urediti po pomembnosti in upoštevanju tveganja izvedbe. Inkrementalni modul prednostno omogoča razvoj osrednjih funkcionalnosti IS. Potrebe po arhitekturnih in vsebinskih spremembah so tako lahko zaznane že v zgodnji fazi celotnega življenjskega cikla. Inkrementalni model je primeren za projekte, kjer mora biti osnovna verzija rešitve hitro implementirana ter predana stranki.

Prednosti inkrementalnega modela:

- Omogoča hitrejši odziv uporabnika, saj so posamezni moduli končne rešitve postopoma predani stranki.
- Lažje obvladovanje tveganj, saj so moduli s prepoznanim večjim tveganjem obravnavani v zgodnejših inkrementih.

- Posamezni moduli IS so bolj učinkovito testirani.
- Omogoča dobro sledenje napredku projekta.

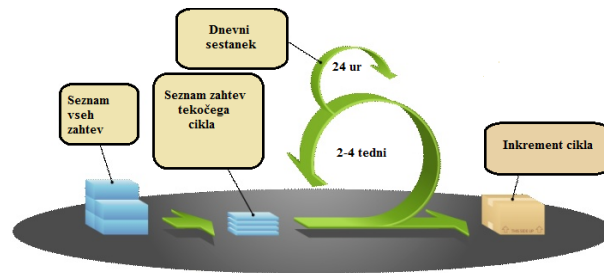
Slabosti inkrementalnega modela:

- Potrebna je dobra analiza in načrtovanje.
- Specifikacija zahtev mora biti dobro definirana, da omogoča razbitje celote na posamezne sklope.
- Vsi projekti ne omogočajo predajanja inkrementov.
- Zapletene integracije med moduli končne rešitve.

### 3.4 Agilne metodologije

Leta 2001 se je 17 izkušenih inženirjev iz področja razvoja IS sestalo v smučarskem središču Snowbird v Utah in postavilo temelje agilne metodologije. Ustanovili so organizacijo Agile Alliance [11], katera širi principe in postopke agilne metodologije. Agilni model postavlja v ospredje posameznika, sodelovanje razvojne skupine in naročnika, delujočo programsko opremo in je hitro prilagodljiv spremembam. Agilne metodologije so večinoma primerne za manjše IS, kjer so razvojne skupine relativno majhne. Implementacija novih zahtev ne povzroča velikega dodatnega napora in ima manjši časovni in finančni obseg. Uspešnost agilne metodologije merimo s kvaliteto dobljenega končnega izdelka. Član ustanovitvenega odbora Agile Alliance Martin Flower navaja, da je rezultat uspešnega agilnega projekta boljši od predvidenega planiranega načrta. Vodenje razvojne skupine ne poteka centralizirano in člani razvojne skupine imajo enakovredne vloge. Ker je agilna metodologija bolj usmerjena k ljudem kot proces, je pomembno, da so člani razvojne skupine visoko motivirani, imajo izkušnje in znanja ter med seboj dobro sodelujejo.

Agilne metodologije temeljijo na štirih načelih [9]:



**Slika 3.4:** Primer agilne metodologije - SCRUM

- Ljudje so pomembnejši od procesov, orodij in tehnik.
- Merilo uspešnosti in glavni cilj je delujoča programska oprema in ne popolna dokumentacija.
- Komunikacija med naročnikom in izvajalcem je aktivna med celotnim projektom.
- Upoštevanje sprememb ima prednost pred sledenjem planu.

Prednosti agilne metodologije:

- Hitro in periodično dobavljanje uporabne programske opreme vpliva na zadovoljstvo stranke.
- Učinkovit mehanizem za prilagajanje spreminjajočim se okoliščinam.
- Dnevna direktna komunikacija med poslovnimi ljudmi in razvojno skupino.
- Med kupci, razvijalci in testerji ves čas poteka interakcija.
- Stalna skrb za tehnično dovršeno programsko opremo.

Slabosti agilne metodologije:

- Ne zagotavlja potrebne dokumentacije.
- Zahteva izkušeno razvojno skupino, kjer so posamezniki sposobni sprejemati vrsto ukrepov.



- 
- Ker zahteve naročnika niso povsem jasne ob začetku projekta, je težko predvideti končni rezultat.
  - Projekt lahko izgubi smernico, saj ga nove zahteve lahko oddaljijo od končnega cilja.
  - Težko je predvideti časovno in stroškovno kompleksnost rešitve.



## Poglavje 4

# Model vrednotenja socio-tehničnih vidikov metodologije razvoja programske opreme

Metodologije razvoja programske opreme obstajajo že nekaj desetletij, a kljub temu mnoga podjetja za razvoj programskih rešitev še vedno ne uporabljajo formalne metodologije za razvoj programske opreme. V organizacijah, kjer ne sledijo formalno predpisani metodologiji, se vseeno uveljavijo procesi, ki so sprejeti na nivoju razvojnih skupin. Uveljavijo se tako imenovani neformalni dogovori med programerji, ki usmerjajo proces samega razvoja. Različni vzroki so zaznani, zakaj razvojne skupine zavračajo in ne sprejmejo formalne metodologije. Pogosti vzrok so ravno neprilagojene metodologije specifični organizaciji in njenim potrebam. Izvajanje in sledenje predpisanim smernicam je lahko premalo prožno in prilagojeno potrebam posameznega projekta. Prav tako so lahko vrednote in sociološke karakteristike članov razvojne skupine preveč različne od predvidenih lastnosti, ki jih metodologija predvideva. Na stopnjo sprejetosti formalne metodologije tako vpliva veliko dejavnikov. Cilj vsake organizacije bi moral biti, da uveljavi in sledi

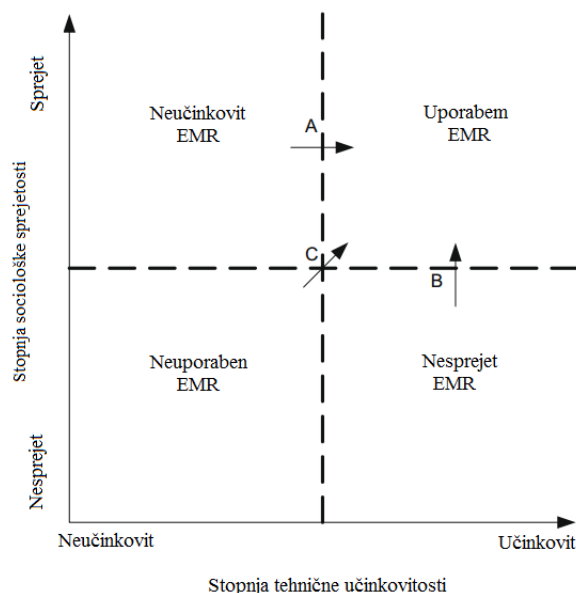
metodologiji, ki je tehnično ustrezna in sociološko sprejeta med člani skupine. Pomemben dejavnik pa je seveda tudi ekonomska vzdržnost izvajanja same metodologije.

Dobro definiran proces izvajanja razvoja programske opreme nam je v veliko pomoč pri doseganju večje učinkovitosti in kakovosti. Če so procesi dobro definirani ter vključeni v metodologijo, je zastavljeni cilj po večji učinkovitosti in kakovosti transparentno dosegljiv. Na metodologijo lahko gledamo kot na skupino posameznih elementov, ki so med seboj močno povezani in preprejeni med uporabniki, ki so člani predpisanih skupin z dobro definiranimi vlogami. Tak način vpogleda nam omogoča, da se pri raziskovanju osredotočimo na posamezne EMR, kot smo jih opredelili v poglavju 2. Z besedo EMR bomo označevali glavne aktivnosti, vloge, priporočila, smernice, predloge, ki jih lahko formalno opredelimo in se v opazovani organizaciji najpogosteje uporabljajo znotraj metodologije. Posamezne EMR želimo z modelom ovrednotiti iz tehničnih in socioloških dimenzij. Ustreznost sociološke dimenzije EMR analiziramo na podlagi socioloških in kulturnih karakteristik razvojne skupine in določimo stopnjo sprejetosti EMR. Ustreznost tehnične dimenzije EMR analiziramo na podlagi tehničnih karakteristik projekta, kot tudi organizacije, in določimo stopnjo učinkovitosti EMR. Namen modela je pomagati organizaciji zaznati manj sprejemljive EMR, katere bi bilo potrebno izboljšati ali morda celo opustiti in zamenjati z boljšimi.

## 4.1 Ogrodje vrednotenja metodologije razvoja

Ogrodje modela za vrednotenje metodologije se osredotoči na preučevanje medsebojnih vplivov socioloških in tehničnih vidikov EMR [7]. Diagram modela je prikazan na sliki 4.1 in je razdeljen na 4 kvadrante. Vsak kvadrant predstavlja svoj tip EMR glede na njihovo vrednost:

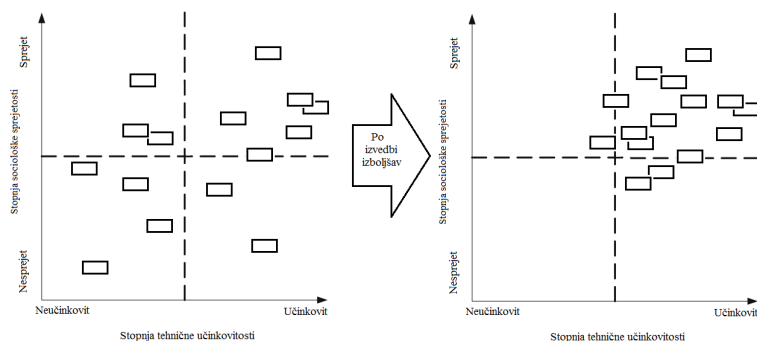
- Neuporaben EMR je sociološko in tehnično neustrezen. Element je lahko imel že od samega začetka neustrezne lastnosti in se zato nikoli v praksi ni uporabljal. Lahko pa je skozi čas in hiter tehnični razvoj



**Slika 4.1:** Ogrodje vrednotenja metodologije razvoja

postal tehnično neustrezen. Element, ki se skozi razvoj ni prilagajal novim tehnologijam in so ga razvojne skupine nehale uporabljati, je prav tako neuporaben.

- Neučinkovit EMR je sociološko sprejet, vendar tehnično neustrezen. Element s takšnimi lastnostmi lahko zaznamo v razvojni skupini, ki je imela v preteklosti že dobro izkustvo s takšnim elementom na predhodnjih projektih. Razvojna skupina je na takšen element navajena in ga tudi uporablja. Tehnično pa je element ovrednoten kot neustrezen, saj novi projekt zahteva drugačne tehnične pristope, kot predhodnji.
- Nesprejet EMR je tehnično ustrezen, vendar sociološko nesprejet. Razvojna skupina v takšnem elementu ne vidi dodane pozitivne vrednosti, zato ga ne uporablja ali pa je stopnja uporabe zelo majhna. Preveč kompleksni elementi so pogosto tehnično zelo dovršeni, a med skupino nesprejeti. Na sociološko nesprejetost prav tako vplivajo tudi izkušnje



**Slika 4.2:** Ogrodje vrednotenja metodologije razvoja

in znanje posameznih članov razvojne skupine.

- Uporaben EMR je sociološko in tehnično ustrezen. Takšni elementi so prilagojeni potrebam razvojnih skupin in projektov. Prinašajo dober rezultat in so osredotočeni na doseganje zastavljenih ciljev.

Namen modela je pomagati organizaciji pri zaznavanju neuporabnih, ne-učinkovitih in nesprejetih EMR, katere bi bilo potrebno nadgraditi, izboljšati ali celo opustiti. Razvrščanje elementov v ogrodje modela raziskovalcu olajša pripravo potrebnih izboljšav za manj ustrezne elemente. Predlagane izboljšave metodologije morajo biti prilagojene danim okoliščinam in so predvsem odvisne od same organizacije, tukaj mislimo na lastnosti razvojnih skupin in narave projektov. Skupno predlogom je, da želimo tehnično dimenzijo dvigniti iz neustrezne v ustrezno, sociološko dimenzijo pa iz nesprejete v sprejeto. Po izvedenih izboljšavah pričakujemo, da bo večina elementov postala tehnično ustreznih in sociološko sprejetih, tako imenovani uporabni EMR, kar prikazuje slika 4.2.

Primer predlogov izboljšav za vsak tip neustreznih EMR.

- Elementu, katerega smo ovednotili kot nesprejetega, je potrebno nadgraditi sociološko dimenzijo. Takšen EMR je v razvojni skupini lahko izdelava tehnične dokumentacije. Tehnično je element ustrezen, saj so smernice s predlogo jasno definirane. Portal, kjer se dokumentacijo

odlaga, je tudi ustrezno urejen. Razvojna skupina pa vseeno izdelavo tehnične dokumentacije zavrača. Člani razvojne skupine so bili ves čas vključeni v razvoj na novih projektih in niso imeli izkustva z uporabo tehnične dokumentacije v fazi podpore živemu projektu. Predlog izboljšave je v prepletanju razvojnih in podpornih skupin. Dodatno znanje in izkušnje s kasnejšim upravljanjem s projektom bi članom razvojne skupine na projektu spremenilo sociološki vidik do izdelave tehnične dokumentacije.

- Elementu, katerega smo ovrednotili kot neučinkovitega, je potrebno nadgraditi tehnično dimenzijo. Primer takšnega EMR je prenos nove verzije programa na produkcijsko okolje. Sociološko je EMR sprejet, saj z novejšo verzijo programa izboljšamo rešitev in člani skupine na ta način zaključijo svoje manjše celote dodelav. Tehnično pa je lahko postopek prenosa nove verzije programa neustrezen. Ni definiranih jasnih postopkov, verjetnost uporabniških napak v postopku prenosa se poveča. Z dodatnimi orodji, kot so na primer skripte, ki se izvajajo po predvidenih urnikih, lahko element nadgradnje programa nadgradimo v uporaben EMR.
- Elementu, katerega smo ovrednotili kot neuporabnega, je potrebno nadgraditi tehnično in sociološko dimenzijo. Če element ni ključnega pomena, ga je najbolje v celoti zavreči. V kolikor pa je element del pomembnega procesa, ga je potrebno zamenjati z bolj ustreznim EMR.

Pri izboljšavah obstajajo omejitve. Vsakega elementa se ne da izboljšati, oziroma nadomestiti s primernejšim. Elementi, ki sestavljajo osrednji proces, so kot takšni ponavadi nenadomestljivi. Že pri prilagajanju dimenzije v bolj sociološko sprejeto in tehnično ustrezno je potrebna večja pazljivost, da se ne pokvari vsebinsko bistvo samega elementa.

## 4.2 Lastnosti sociološke sprejetosti

Stanje sociološke ustreznosti merimo z dvema lastnostima.

- S pogostostjo uporabe EMR ob predpostavki, da so pogoji ustrezni za uporabo EMR, merimo, kako pogosto se člani razvojne skupine odločijo za uporabo posameznega EMR. Vseh EMR znotraj opazovane metodologije se ne uporablja enako pogosto, zato merimo pogostost uporabe glede na relativno možnost uporabe posameznega EMR. Model za vrednotenje metodologije nam omogoča takšen pristop, saj ne merimo lastnosti na nivoju celotne metodologije, temveč na nivoju posameznih EMR.
- Z doslednostjo uporabe EMR merimo, kako dosledno razvojna skupina izvaja navodila in pravila, predpisana za uporabo posameznega EMR. V opazovanem okolju želimo preveriti, kako so pravila zaživila na projektnem delu.

Razloge za sociološko ustreznost merimo z množico lastnosti, ki jih vsebinsko razdelimo v dve podmnožici. Pri večini lastnosti se osredotočimo na uporabniško dožemanje EMR znotraj organizacije.

- Množica lastnosti, s katerimi merimo stopnjo dožemanja EMR pri uporabnikih metodologije.
  - Z relativno prednostjo merimo stopnjo dožemanja uporabnosti EMR pri uporabniku. Ta lastnost najpomembneje vpliva na stopnjo sprejetosti EMR. Lastnost merimo relativno, saj nas zanima, kako uporabnik dožema uporabnost v primerjavi z neuporabo EMR.
  - S sociološko združljivostjo merimo stopnjo dožemanja EMR pri uporabniku z njegovim predhodnim znanjem, izkušnjami in potrebami.



- S kompleksnostjo merimo stopnjo dojemanja EMR, kako težko je uporabljati sam element. Prevelika kompleksnost izrazito negativno vpliva na sprejetost elementa.
- Množica lastnosti, s katerimi merimo lastnosti uporabnikov metodologije.
  - Uporabniška znanja in izkušnje na področju razvoja programske opreme. Bolj izkušeni uporabniki, ki imajo več znanj o tehnologiji, razvojnih okoljih, programskih orodjih in programskih jezikih, imajo večje zadržke in težje sprejmejo EMR. Pri uporabniku ovrednotimo znanje in izkušnje, ki se direktno navezujejo na ovrednoteni EMR.
  - Uporabniška znanja in izkušnje na področju metodologij razvoja programske opreme. Uporabniki, pri katerih zaznamo več izkušenj in znanj s področja procesa razvijanja programske opreme, projektnega vodenja in standardov, bolje sprejemajo EMR. Pri uporabniku ovrednotimo znanje in izkušnje, ki se direktno navezujejo na ovrednoteni EMR.

### 4.3 Lastnosti tehnične učinkovitosti

Stanje tehnične učinkovitosti merimo s 4 lastnostmi.

- S pogostostjo priložnosti za uporabo EMR merimo, kako pogosto se člani razvojne skupine odločijo za uporabo posameznega EMR, ne glede na to, ali se opazovani EMR uporablja med člani.
- Skupina lastnosti, s katerimi merimo vključenost EMR v sistem, ki ga razvojna skupina razvija. Lastnosti so: skladnost, popolnost, uporabnost, zanesljivost, učinkovitost, prenosljivost, vzdržljivost, ponovna uporabnost.

- Skupina lastnosti, s katerimi merimo vključenost EMR v projektu. Lastnosti so: porabljenost časa, stroški projekta, nadzor nad projektom, kakovost projektnega plana, sledljivost projektu, ocenjevanje tveganj in ostalih problemov znotraj projekta.
- Skupina lastnosti, s katerimi merimo vključenost EMR med uporabnike EMR. Lastnosti so: dvoumnost komunikacije, olajšanje sodelovanja, podpiranje izboraževanja, razumevanje dolžnosti in odgovornosti.

Razloge za tehnično učinkovitost merimo z množico lastnosti, katere vsebinsko razdelimo v dve podmnožici.

- Množica lastnosti, s katerimi merimo tehnično ustreznost, vsebuje tri lastnosti.
  - Ustreznost projektu in razvojnemu sistemu je stopnja, do katere uporabniki razvojne skupine zaznavajo EMR kot ustrezen različnim projektom in različno velikim, kompleksnim sistemom.
  - Ustreznost razvojni skupini je mera, s katero iz objektivne perspektive ocenimo ustreznost EMR glede na znanje in izkušnje članov razvojne skupine.
  - Skladnost z modernimi razvojnimi pristopi je stopnja, kjer merimo, kako EMR sledi aktualnim trendom in novostim na področju tehnologije.
- Množica lastnosti, s katerimi merimo tehnično skladnost, vsebuje tri lastnosti.
  - Skladnost z razvojnim okoljem in informacijskimi tehnologijami je stopnja, do katere EMR ustreza tehnologiji in programskemu okolju pri razvoju sistema na projektu.
  - Skladnost z internimi normami je stopnja, do katere EMR izpolnjuje notranje standarde metodologije in skupne standarde z ostalimi EMR znotraj metodologije. Z lastnostjo merimo notranjo integriteto EMR.

- 
- Skladnost s splošnimi normami je stopnja, do katere opazovani EMR izpolnjuje splošne definirane standarde, kot so arhitekturni standardi, standardi za podatkovno modeliranje, za dokumentiranje, verzioniranje in kodiranje.



## Poglavje 5

# Vrednotenje metodologije znotraj opazovane razvojne skupine

Praktični del diplomskega dela smo izvedli znotraj večjega slovenskega računalniškega podjetja, natančneje v razvojni skupini oddelka za implementacijo sistema za upravljanje poslovnih rešitev Microsoft Dynamics AX. Oddelek je sestavljen iz približno 60 strokovnjakov, ki jih v splošnem lahko razdelimo med svetovalce in razvijalce poslovnih rešitev.

Pri analizi stanja se bomo osredotočili na razvijalce, ki so z različnimi vlogami vključeni v razvojne skupine. Razvojne skupine štejejo med 4 do 6 inženirjev, ki imajo zelo različna znanja in izkušnje. Ker se oddelek stalno povečuje, je potreba po večji učinkovitosti in kakovosti ves čas prisotna. Dober odgovor nam lahko poda podrobna analiza obstoječe metodologije in vrednotenje posameznih zaznanih EMR po modelu iz poglavja 4. V raziskavo smo vključili 10 razvojnih inženirjev, ki so člani različnih razvojnih skupin. Opazovali smo vse dejavnosti, ki se odvijajo znotraj skupine, in jih poskušali formalizirati. Ključne EMR smo vključili v merilni instrument in jih po modelu uvrstili v enega izmed 4 tipov EMR. V analizo so bili vključeni visoko motivirani razvojni inženirji, ki so bili skrbno izbrani, s čimer smo zagotovili

pravo ravnovesje med vlogami.

## 5.1 Vloge in naziv članov razvojne skupine

V opazovanem oddelku imajo člani razvojnih skupin jasno definirane nazive, kateri opredeljujejo njihove delovne aktivnosti. Splošne vloge tehnične stroke v razvojni skupini, kot so: programer, načrtovalec, tester, v organizaciji niso ločene v svoje vloge. Za potrebe vrednotenja metodologije smo zato razvojne inženirje razdelili v skupine glede na nazive, da lahko nazive izenačimo z vlogami. Zaposlenemu je dodeljen naziv na podlagi izkušenj in znanj. Člani razvojne skupine lahko sodelujejo v različnih projektih sočasno, vendar se njihova vloga ne spreminja. Raven odgovornosti in množica aktivnosti za katere skrbi, ostaja na vseh projektih enaka. Napredovanje zaposlenega je v podjetju tesno povezano s prehajanjem med nazivi. Potrebna znanja in izkušnje so specificirana v pravilniku, ki določa potrebne in zadostne pogoje za napredovanje. Vloge zaposlenih tako temeljijo na letih delovnih izkušenj ter pridobljenem znanju.

V opazovani skupini smo inženirje razdelili v 4 skupine:

- Mlajši razvijalec programske opreme
- Razvijalec programske opreme
- Izkušeni razvijalec programske opreme
- Arhitekt poslovnih rešitev

Posamezno skupino bomo opisali in jo povezali z množico pripadajočih EMR.

### 5.1.1 Mlajši razvijalec programske opreme

V skupino mlajši razvijalec programske opreme smo uvrstili inženirja, ki nima veliko znanj in izkušenj s produktom Microsoft Dynamics AX in ima med 0 in

Zaznani EMR	tip EMR
Vzpostavitev infrastrukture na razvojnem okolju	Aktivnost
Ocena dodelav	Aktivnost
Razvoj novih funkcionalnosti in dodelav na obstoječih funkcionalnostih v x++	Aktivnost
Razvoj SSRS poročil	Aktivnost
Preverba kode	Aktivnost
Izobraževanja standardnih funkcionalnosti	Aktivnost
Sestanki v sklopu projektnega vodenja	Aktivnost
Osnutek za oddajo ponudb	Dokument
Jira, orodje za kreiranje in spremljanje zahtevkov	Orodje
TFS, sistem za nadzor različic	Orodje

**Tabela 5.1:** Zaznani EMR v skupini mlajših razvijalcev programske opreme

3 leta delovnih izkušenj. To skupino bomo označili s stopnjo 1 in predstavlja vstopno stopnjo za mladega inženirja, ki je nastopil svojo prvo zaposlitev. Zgornja meja je omejena na 3 leta, saj se pričakuje, da se zahtevane veščine osvoji v predpisanem obdobju. Številčno je ta skupina zelo majhna, saj je interes njenih članov po napredovanju izjemno velik. V našo raziskavo smo zato vključili le 1 člana iz omenjene skupine.

Mlajši razvijalec programske opreme se je sposoben osredotočiti na rok izvedbe in je motiviran, da doseže dober končni rezultat. Njegova komunikacija s preostalimi člani razvojne skupine je jasna in učinkovita.

V tabeli 5.1 so predstavljeni EMR, ki jih mladi razvijalci programske opreme uporabljajo in so odgovorni za njihovo izvedbo.

### 5.1.2 Razvijalec programske opreme

Razvijalec programske opreme ima tehnična in funkcionalna znanja o Microsoft Dynamics AX in pozna razvojno tehnologijo in okolje. To skupino

Zaznani EMR	tip EMR
Vzpostavitev infrastrukture na razvojnem okolju	Aktivnost
Analiza in zajem zahtev	Aktivnost
Ocena dodelav	Aktivnost
Načrtovanje rešitve	Aktivnost
Razvoj novih funkcionalnosti in dodelav na obstoječih funkcionalnostih v x++	Aktivnost
Razvoj SSRS poročil	Aktivnost
Preverba kode	Aktivnost
Testiranje	Aktivnost
Izobraževanja standardnih funkcionalnosti	Aktivnost
Sestanki v sklopu projektnega vodenja	Aktivnost
Osnutek FRD, DoR in DoD	Dokument
Osnutek za oddajo ponudb	Dokument
Tehnična dokumentacija	Dokument
TFS, sistem za nadzor različic	Orodje
Jira, orodje za kreiranje in spremljanje zahtevkov	Orodje

**Tabela 5.2:** Zaznani EMR v skupini razvijalcev programske opreme

smo označili s stopnjo 2 in zahteva med 2 in 5 let delovnih izkušenj. V raziskavo smo vključili 3 člane razvijalcev programske opreme. Razvijalec programske opreme je sposoben abstraktnega razmišljanja, deluje samostojno in zna reševati probleme proaktivno. Ta skupina je znotraj opazovanega oddelka najštevilčnejša. Velik poudarek je na dobrem sodelovanju, močni komunikacijski spretnosti in osredotočenosti na kupca. Zagotavlja celovito podporo in strokovno znanje na področju razvoja v vseh fazah projekta, še posebej pa v fazi izvedbe in testiranja.

V tabeli 5.2 so predstavljeni EMR, ki jih razvijalci programske opreme uporabljajo in so odgovorni za njihovo izvedbo.



### 5.1.3 Izkušeni razvijalec programske opreme

Izkušeni razvijalec programske opreme ima poglobljena znanja o produktu, katera je pridobil vsaj 6 let. Zaradi bogatih izkušenj, ki jih je pridobil na različnih projektih več let, ima v organizaciji pomembno vlogo mentorstva. Sposoben je reševati najzahtevnejše probleme, saj pozna celotno arhitekturno zasnovo produkta Microsoft Dynamics AX. Prav zaradi te lastnosti je nepogrešljiv v fazi testiranja, kjer ima popoln pregled nad vplivom delav in novih funkcionalnosti na obstoječo standardno rešitev. Prisotni so v vseh fazah življenjskega cikla, pomembnejšo vlogo pa imajo v fazi analize in načrtovanja, pa tudi pri uvedbi in migraciji podatkov. Ves čas strmi k učinkovitemu delovanju in izboljšavam ter uvedbam dobrih praks. V raziskavo smo vključili 4 izkušene razvijalce programske opreme in to skupino označili s stopnjo 3.

V tabeli 5.3 so predstavljeni EMR, ki jih izkušeni razvijalci programske opreme uporabljajo in so odgovorni za njihovo izvedbo.

Zaznani EMR	tip EMR
Vzpostavitev infrastrukture na razvojnem okolju	Aktivnost
Analiza in zajem zahtev	Aktivnost
Ocena dodelav	Aktivnost
Načrtovanje rešitve	Aktivnost
Razvoj novih funkcionalnosti in dodelav na obstoječih funkcionalnostih v x++	Aktivnost
Razvoj SSRS poročil	Aktivnost
Preverba kode	Aktivnost
Testiranje	Aktivnost
Izobraževanja standardnih funkcionalnosti	Aktivnost
Sestanki v sklopu projektnega vodenja	Aktivnost
Razvoj OLAP	Aktivnost
Razvoj in implementacija integracij	Aktivnost
Performančna analiza	Aktivnost
Prenos nove verzije produkta na testno okolje, generiranje build	Aktivnost
Prenos nove verzije produkta na produkcijsko okolje	Aktivnost
Osnutek FRD, DoR in DoD	Dokument
Osnutek za oddajo ponudb	Dokument
Tehnična dokumentacija	Dokument
Protokol prenosa nove verzije produkta	Dokument
TFS, sistem za nadzor različic	Orodje
Jira, orodje za kreiranje in spremljanje zahtevkov	Orodje
DMF,Atlas, orodja za migracijo podatkov	Orodje

**Tabela 5.3:** Zaznani EMR v skupini izkušenih razvijalcev programske opreme

#### 5.1.4 Arhitekt poslovnih rešitev

Arhitekt poslovnih rešitev zagotavlja tehnično kot tudi vsebinsko znanje o izdelku Microsoft Dynamics AX na najvišji ravni in je strokovnjak tudi za ostale povezane Microsoft tehnologije. Je aktivno vključen v fazi analize in načrtovanja. Določa potrebne funkcionalnosti, predstavi primere uporabe, vključen je pri definicijah vmesnikov in obremenitvenih scenarijih. Postavi temelj arhitekturnega koncepta rešitve skladno z zahtevami in omejitvami kupca. Je ključna oseba pri postavitvi strategije integracij novega ERP-sistema z ostalimi IS v podjetju. Arhitekti poslovnih rešitev sodelujejo v organizaciji z vodstvenim kadrom in imajo dobre sposobnosti načrtovanja in vodenja projekta. Izvajajo podatkovno in procesno modeliranje ter analitično analizirajo in razumejo poslovni proces kupca. V organizaciji imajo strateški pomen in njihove dolgoletne izkušnje vplivajo na metodologijo v organizaciji. V raziskavo smo vključili 2 arhitekta poslovnih rešitev in to skupino označili s stopnjo 4.

V tabeli 5.4 so predstavljeni EMR, ki jih arhitekti poslovnih rešitev uporabljajo in so odgovorni za njihovo izvedbo. V tabeli 5.4 smo EMR označili še z oznakami, ki smo jih uvrstili v procesni diagram na sliki 5.1 in v razsevni diagram na sliki 6.2.

Zaznani EMR	Tip EMR	Oznaka
Vzpostavitev infrastrukture na razvojnem okolju	Aktivnost	Q14
Vzpostavitev infrastrukture na produkcijskem okolju	Aktivnost	Q15
Analiza in zajem zahtev	Aktivnost	Q1
Ocena dodelav	Aktivnost	Q3
Načrtovanje rešitve	Aktivnost	Q6
Razvoj novih funkcionalnosti in dodelav na obstoječih funkcionalnostih v x++	Aktivnost	Q8
Razvoj SSRS poročil	Aktivnost	Q9
Preverba kode	Aktivnost	Q12
Testiranje	Aktivnost	Q13
Izobraževanja standardnih funkcionalnosti	Aktivnost	Q21
Sestanki v sklopu projektnega vodenja	Aktivnost	Q23
Razvoj OLAP	Aktivnost	Q10
Razvoj in implementacija integracij	Aktivnost	Q11
Performančna analiza	Aktivnost	Q22
Prenos nove verzije produkta na testno okolje, generiranje build	Aktivnost	Q16
Prenos nove verzije produkta na produkcijsko okolje	Aktivnost	Q17
Nadgradnja verzije MS AX in lokalizacije	Aktivnost	Q24
Osnutek FRD, DoR in DoD	Dokument	Q2
Osnutek za oddajo ponudb	Dokument	Q4
Tehnična dokumentacija	Dokument	Q7
Protokol prenosa nove verzije produkta	Dokument	Q18
TFS, sistem za nadzor različic	Orodje	Q19
Jira, orodje za kreiranje in spremljanje zahtevkov	Orodje	Q5
DMF,Atlas, orodja za migracijo podatkov	Orodje	Q20

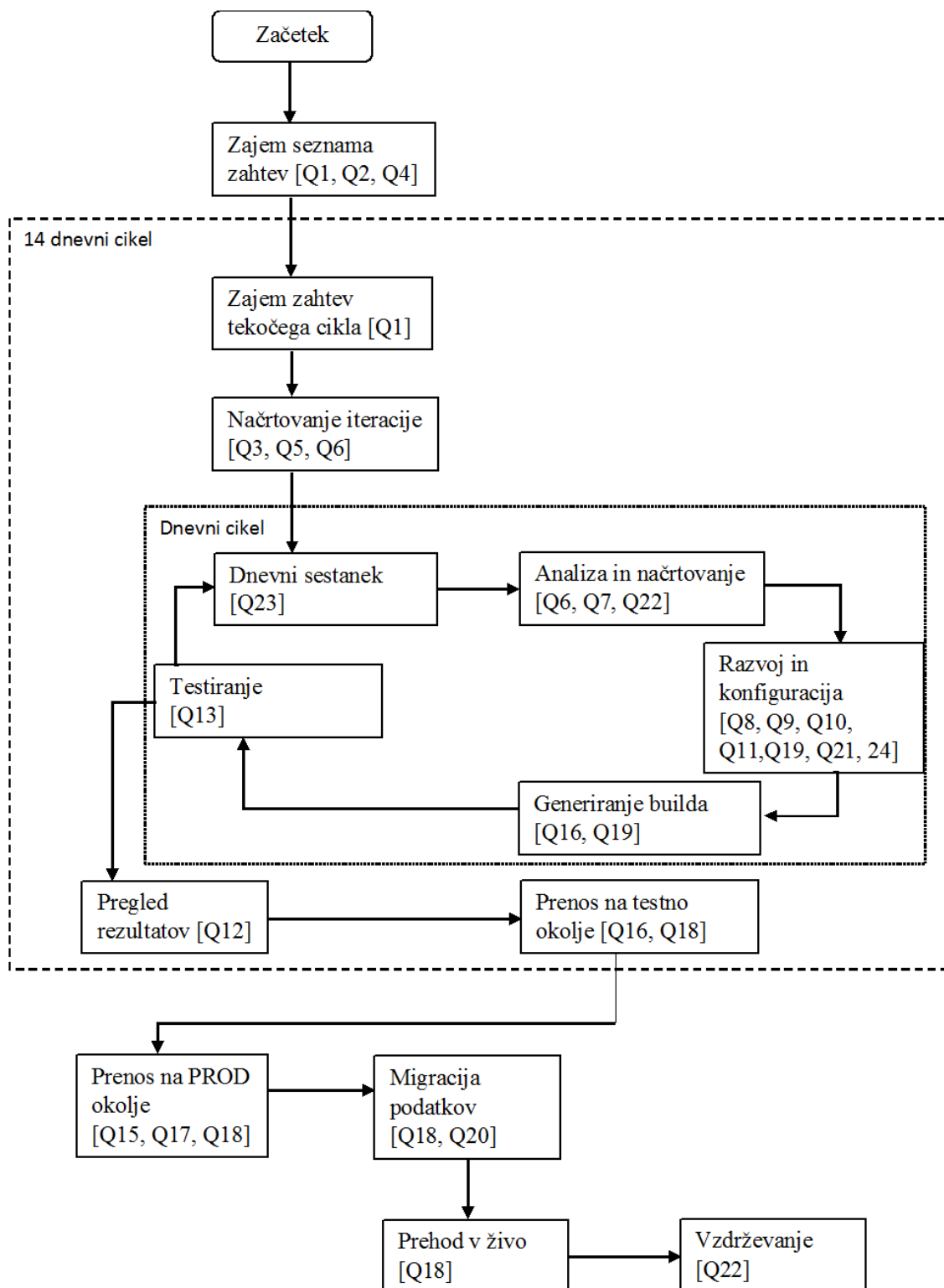
**Tabela 5.4:** EMR zaznani v skupini arhitektov poslovnih rešitev

## 5.2 Procesni diagram poteka

Želeli smo ugotoviti, kateri poslovni procesi potekajo v razvojni skupini, zato smo naredili posnetek obstoječega stanja procesov. S pomočjo modeliranja bomo ponazorili ogrožje poslovnega procesa. Predstavili bomo osnovni tok procesov in zaradi kompleksnosti in nepreglednosti ne bomo obravnavali podprocesov v modelu. Z modelom želimo predstaviti realno sliko dogajanja v projektu iz zornega kota inženirja v opazovani razvojni skupini. V ta namen bomo zanemarili za nas nepomembne procese, ki jih inženirji ne izvajajo. Z modelom ne želimo odkriti slabosti v izvajanju procesa, saj to ni naš namen. Želimo pa predstaviti boljšo sliko poslovnega procesa v projektnem delu pri implementaciji MS Dynamics AX, v katerem smo zaznali 24 ključnih in formalno opredeljenih EMR. Pri izbiri elementov za evalvacijo smo si pomagali s procesnim modelom, kjer smo za posamezni proces poiskali vsebovane formalno opredeljene aktivnosti, uporabljena orodja ter predpisane dokumente.

S sliko 5.1 smo grafično predstavili proces, kjer smo uporabili tehniko procesnih diagramov. Pri posameznem procesu smo zapisali seznam vključenih EMR.

Opazovana razvojna skupina uporablja prilagojeno iterativno in inkrementalno agilno metodologijo Scrum, zato se velik del aktivnosti iterativno izvaja znotraj 14-dnevnega cikla. Ob začetku novega cikla se izvede aktivnost zajema zahtev iz celotnega popisanega seznama zahtev in se naredi načrt tekoče iteracije. Znotraj 14-dnevnega cikla pa se odvijajo aktivnosti na dnevnem nivoju. Vsak dan se tako izvede dnevni sestanek, kjer se preveri delo preteklega dne in predstavi plan dela za tekoči dan. Tekom dneva inženirji izvajajo aktivnosti analize in načrtovanja rešitve, razvoj in konfiguracijo produkta ter proces testiranja. Po končanem 14-dnevnem ciklu se izvede pregled rezultatov in, v kolikor so pričakovani rezultati primerni, se izvede prenos novega inkrementa na testno okolje pri stranki. Po potrditvi pravilnosti, se s stranko uskladi in prenese novo verzijo produkta na produkcijsko okolje. Inženirji so vključeni tudi v proces migracije podatkov, prehoda produkta v živo ter v zadnjo fazo vzdrževanja.



Slika 5.1: Model procesa

### 5.3 Merilni inštrument

Za potrebe vrednotenja EMR smo kreirali merilni inštrument po usmeritvah modela za socio-tehnično vrednotenje metodologije, predstavljenega v 4. poglavju. Merilni inštrument je vključeval povezave med skupinami in EMR, tako da je vsak član razvojne skupine odgovarjal na vprašanja, povezana le z EMR, za katere je odgovoren. Za vsak tip EMR smo postavili enak nabor trditev, na katere je bilo potrebno odgovoriti s stopnjo strinjanja. Izkušeni razvijalci programske opreme in arhitekti poslovnih rešitev so odgovarjali na vse trditve, mlajšim razvijalcem in razvijalcem programske opreme pa smo postavili le prve štiri zaporedne trditve iz celotnega seznama trditev za posamezni tip EMR, ki se navezujejo na sociološko dimenzijo. Na lestvici od 1 do 7, pri čemer 1 pomeni, sploh se ne strinjam, 7 pa, popolnoma se strinjam, so člani razvojnih skupin ocenili aktivnosti, orodja ter dokumente.

Merilni inštrument sestavljajo trditve o EMR, ki smo jih v tabeli 5.5 združili v skupine, katere povprašujejo po skupni lastnosti.

Trditve	Oznaka
Aktivnost je združljiva z mojimi izkušnjami in znanjem Dokument je združljiv z mojimi izkušnjami in znanjem Orodje je združljivo z mojimi izkušnjami in znanjem	A
Zadovoljen sem z načinom izvajanja aktivnosti Zadovoljen sem z vsebino in obliko dokumenta Zadovoljen sem z funkcionalnostjo orodja	B
Aktivnost imam za sprejeto in za običajno aktivnost Dokument imam za sprejetega in za del običajnega procesa Orodje imam za sprejeto in za del običajnega procesa	C
Kako pogosto izvajam aktivnost Kako pogosto sem v stiku z dokumentom Kako pogosto sem v stiku z orodjem	D
Aktivnost pripomore k boljši učinkovitosti razvojne skupine Dokument pripomore k boljši učinkovitosti razvojne skupine Orodje pripomore k boljši učinkovitosti razvojne skupine	E
Aktivnost je primerna za delo na AX projektih Dokument je primeren za delo na AX projektih Orodje je primerno za delo na AX projektih	F
Aktivnost je skladna z modernimi razvojnimi pristopi Dokument je skladen z modernimi razvojnimi pristopi Orodje je skladno z modernimi razvojnimi pristopi	G

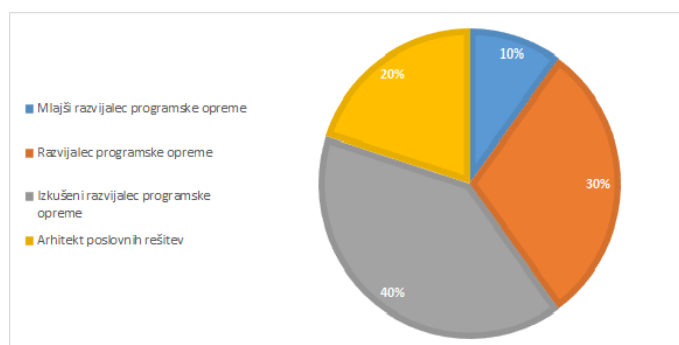
**Tabela 5.5:** Trditve merilnega inštrumenta, združene v skupine z oznakami



## Poglavje 6

### Analiza rezultatov

V raziskavo smo vključili 10 inženirjev, ki smo jih glede na nazive razdelili v različne skupine. Razmerje med posameznimi skupinami smo grafično prikazali na sliki 6.1. Vključeni inženirji so odgovarjali zelo dosledno, zato smo dobili vse vprašalnike v celoti izpolnjene. Število vprašanj in število različnih EMR, na katere je odgovarjal član razvojne skupine, je bilo različno, glede na njegovo vlogo. Na 168 vprašanj so odgovorili arhitekti poslovnih rešitev, mlajšim razvijalcem programske opreme pa smo zastavili 40 vprašanj. V povprečju so člani razvojne skupine potrebovali med 10 in 20 minut za izpolnitev vprašalnika.

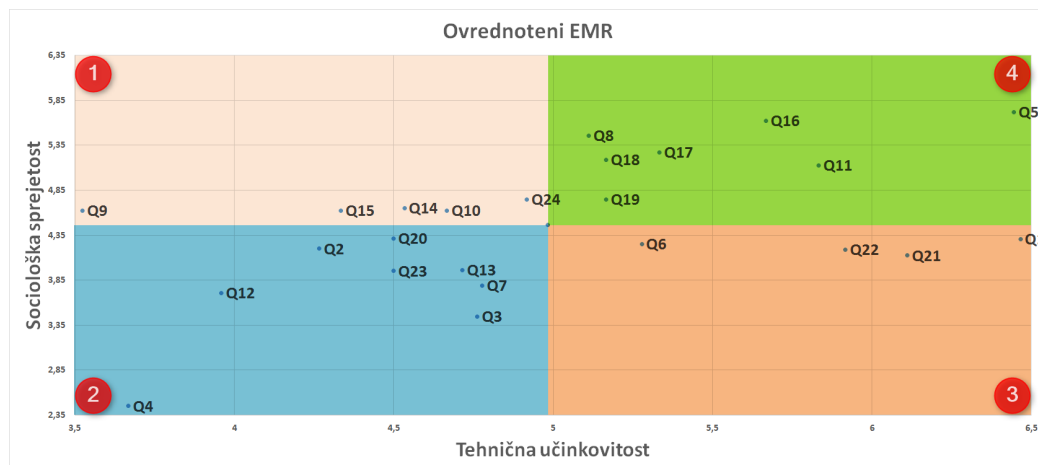


**Slika 6.1:** Razmerja med skupinami vključenih inženirjev

## 6.1 Prikaz rezultatov vrednotenja socio-tehničnih dimenzij za vse elemente

Skupni izračun povprečne vrednosti odgovorov na trditve iz skupin A, B, C in D, predstavljenih v poglavju 5.3, je vsakemu EMR določal vrednost na y osi, kjer smo predstavili sociološko sprejetost. Skupni izračun povprečne vrednosti odgovorov iz skupin E, F in G, predstavljenih v poglavju 5.3, je vsakemu EMR določal vrednost na x osi, kjer smo predstavili tehnično učinkovitost.

Za vsako dimenzijo smo izračunali povprečno vrednost ter na podlagi dobljenih rezultatov oblikovali kvadrante. Mejo sociološke sprejetosti smo postavili na stopnjo strinjanja 4.4, mejo tehnične ustreznosti pa smo postavili na stopnjo strinjanja 4.9. Povzamemo lahko, da so anketiranci podali svoje odgovore na pozitivnem delu skale. Ker nas ne zanimajo absolutne vrednosti za posamezni EMR, temveč razmerja med vsemi EMR, smo s takšnim načinom oblikovanja kvadrantov dobili bolj uravnotežene skupine EMR.



**Slika 6.2:** Ovrednoteni EMR

Na sliki 6.2 smo predstavili rezultate evalvacije na razsevnem diagramu. Povezavo med EMR in njegovo oznako smo predstavili v tabeli 5.4.

Sliko 6.2 smo razdelili na 4 kvadrante in jih označili z zaporedno številko. Za vsak kvadrant bomo našli seznam pripadajočih EMR.

- Kvadrant z zaporedno številko 1 je ogrodje za neučinkovite EMR.

Neučinkoviti EMR in njihove oznake:

- Aktivnost razvoj SSRS poročil (Q9)
- Aktivnost razvoj OLAP (Q10)
- Aktivnost vzpostavitev infrastrukture na produkcijskem okolju (Q15)
- Aktivnost vzpostavitev infrastrukture na razvojnem okolju (Q14)
- Aktivnost nadgradnja verzije Microsoft Dynamics AX in lokalizacije (Q24)

- Kvadrant z zaporedno številko 2 je ogrodje za neuporabne EMR.

Neuporabni EMR in njihove oznake:

- Dokument osnutek za oddajo ponudb (Q4)
  - Aktivnost preverba kode (Q12)
  - Dokument osnutek FRD, DoR in DoD (Q2)
  - Aktivnost ocena dodelav (Q3)
  - Aktivnost testiranje (Q13)
  - Aktivnost sestanki v sklopu projektnega vodenja (Q23)
  - Orodje DMF,Atlas, orodja za migracijo podatkov (Q20)
  - Dokument tehnična dokumentacija (Q7)
- Kvadrant z zaporedno številko 3 je ogrodje za nesprejete EMR.

Nesprejeta EMR in njuni oznaki:

- Aktivnost načrtovanje rešitve (Q6)
  - Aktivnost performančna analiza (Q22)
  - Aktivnost izobraževanja standardnih funkcionalnosti (Q21)
  - Aktivnost analiza in zajem zahtev (Q1)
- Kvadrant z zaporedno številko 4 je ogrodje za uporabne EMR.

Uporabni EMR in njihove oznake:

- Orodje Jira, orodje za kreiranje in spremljanje zahtevkov (Q5)
- Aktivnost razvoj novih funkcionalnosti in dodelav na obstoječih funkcionalnostih v x++ (Q8)
- Aktivnost razvoj in implementacija integracij (Q11)
- Aktivnost prenos nove verzije produkta na testno okolje, generiranje build (Q16)
- Aktivnost prenos nove verzije produkta na produkcijsko okolje (Q17)
- Dokument protokol prenosa nove verzije produkta (Q18)
- Orodje TFS, sistem za nadzor različic (Q19)

## 6.2 Prikaz ovrednotenja posameznega EMR

V poglavju 5.3 smo predstavili nabor različnih vprašanj, ki smo jih zastavili za ERM tipa aktivnost, orodje in dokument. V tem poglavju bomo grafično predstavili raven stopnje strinjanja članov razvojne skupine na zastavljena vprašanja za tri izbrane EMR iz različnih kvadrantov. Izbrane EMR bomo predstavili in grafično prikazali rezultate, dobljene z vprašalnikom o EMR.

Preverili smo stopnjo zadovoljstva, sprejetosti in združljivosti z njihovimi izkušnjami in znanjem ob izvajanju EMR. Zanimal nas je tudi njihov pogled na učinkovitost, skladnost in naprednost EMR. Člani razvojnih skupin niso vedeli, da posamezna vprašanja povezujemo s sociološko sprejetostjo, druga pa s tehnično učinkovitostjo. Pri grafičnem prikazu rezultatov smo pri vseh EMR uporabili enak vrstni red vprašanj. Neodvisno od tipa EMR se zaporedno vprašanje navezuje na skupno lastnost, po kateri smo spraševali, pa čeprav je morda definicija vprašanja prilagojena tipu EMR.

Za podrobnejši opis aktivnosti razvoj SSRS poročil smo se odločili, ker se aktivnost pogosto izvaja dlje od pričakovanega časovnega okvira. Člani razvojnih skupin pogosto izvajajo razvoj SSRS poročil, zato menimo, da ima aktivnost velik potencial za izboljšave. Dobre smernice za natančno oceno potrebnega časa za izvedbo projekta nam predpisuje dokument osnutek za oddajo ponudb. Dokument bomo predstavili, ker ocenjujemo, da ima velik vpliv na pripravo proračuna projekta in med člani razvojnih skupin dokument ni v široki uporabi. Predstavili bomo še aktivnost performančna analiza, saj menimo, da se aktivnost v opazovani organizaciji ne izvaja optimalno in zato potrebuje izboljšave pri izvedbi.

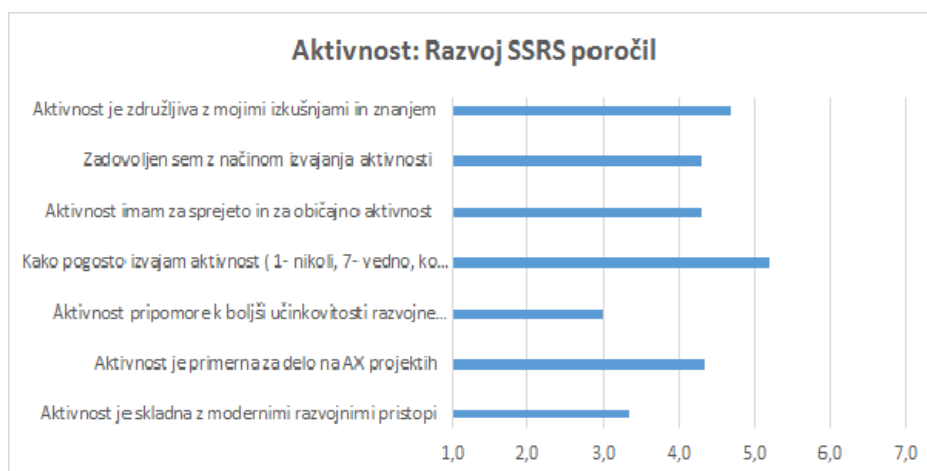
### 6.2.1 Razvoj SSRS poročil (Q9 )

Microsoft SQL Server Reporting Services (SSRS) je primarna platforma za poročanje v Microsoft Dynamics AX. Privzeta in vnaprej definirana poročila, ki so na voljo v Microsoft Dynamics AX, delujejo na Reporting Services tehnologiji. Predpogoj za uporabo in razvoj SSRS poročil je uspešno za-

ključena namestitev Reporting Services, konfiguracija Reporting Services za uporabo z Microsoft Dynamics AX in objava definicij poročil.

Razvojni inženirji izvajajo aktivnost razvoja SSRS poročil v opazovani razvojni skupini zelo pogosto. Za prilagajanje obstoječih poročil, razvoj novih poročil in integracijo teh poročil z Microsoft Dynamics AX uporabljajo orodje Microsoft Visual Studio. Obstaja več metod razvoja SSRS poročil v Microsoft Dynamics AX. Osnovna SSRS poročila, ki ne vsebujejo poslovne logike, uporabljajo poizvedbe, katere so definirane v AOT znotraj Microsoft Dynamics AX. Zahtevnejša poročila pa za dostop in obdelavo podatkov uporabljajo Report Data Provider (RDP) razred.

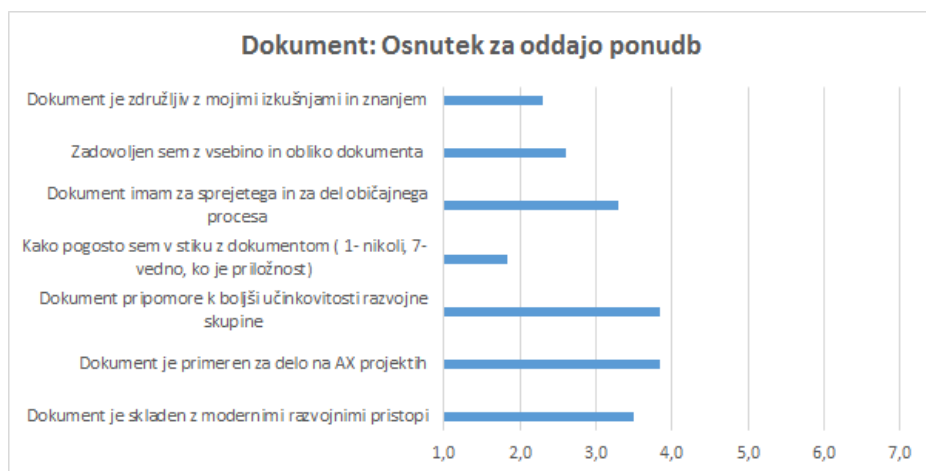
Aktivnost razvoj SSRS poročil smo ovrednotili kot neučinkovit EMR. Zaznali smo, da ima EMR sociološko komponento nad povprečno vrednostjo, saj razvojni inženirji sprejemajo EMR za običajno aktivnost in imajo z razvojem poročil izkušnje in znanje. Tehnična učinkovitost opazovanega EMR pa ima podpovprečno ocenjeno vrednost. Zaradi težav, s katerimi se srečujejo ob razvoju SSRS poročil, ocenjujejo, da tak način razvoja SSRS poročil in integracija z Microsoft Dynamics AX ni skladna z modernimi razvojnimi pristopi. Zato so tudi ocenili, da EMR ne pripomore k večji učinkovitosti razvojne skupine. Pogosto zaznane težave pri razvoju SSRS poročil so: nestabilno razvojno okolje, omejitve pri izgledu poročil in časovne ter prostorske omejitve.

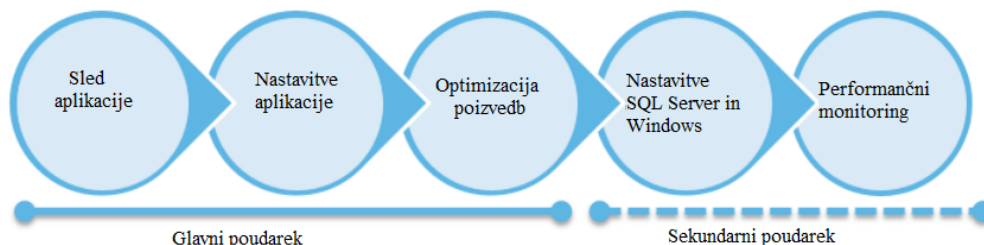


### 6.2.2 Osnutek za oddajo ponudb (Q4 )

Z namenom po boljši sistematičnosti in doslednosti pri ocenah potrebnega časa za izvedbo projekta, je organizacija vpeljala dokument osnutek za oddajo ponudb. Razvojni inženirji ocenijo porabo potrebnega časa za izvedbo aktivnosti na podlagi njihovih izkušenj in znanj. Pomembno je, da dobro poznajo področje dela, ki ga ocenjujejo, ter da niso prisiljeni k oddaji ocene. Ocene potrebnega časa se merijo v porabljenih urah ali dnevih. Opazovani EMR je tesno povezan z EMR za kreiranje in spremljanje zahtevkov. Beleženje porabljenega časa pomaga pridobiti občutek za oceno resnično porabljenega časa. Za boljšo oceno dokument razdeli ocene po različnih kategorijah. Kategorije, ki jih je potrebno izpolniti so: analiza uporabniške zahteve, načrtovanje funkcionalne rešitve, priprava testnega scenarija, tehnično načrtovanje, razvoj (kodiranje), test razvijalca, test svetovalca, tehnično dokumentiranje dodelave , priprava uporabniških navodil, priprava uporabniških pravic, prenos rešitve v produkcijsko okolje, prenos podatkov, nastavitve, izobraževanje, pomoč pri prehodu v živo, projektno vodenje in koordinacija.

Dokument osnutek za oddajo ponudb smo ovrednotili kot neuporaben EMR. Zaznali smo, da ga inženirji zelo redko uporabljajo. Element je ocenjen podpovprečno, tako po sociološki kot tehnični dimenziji.



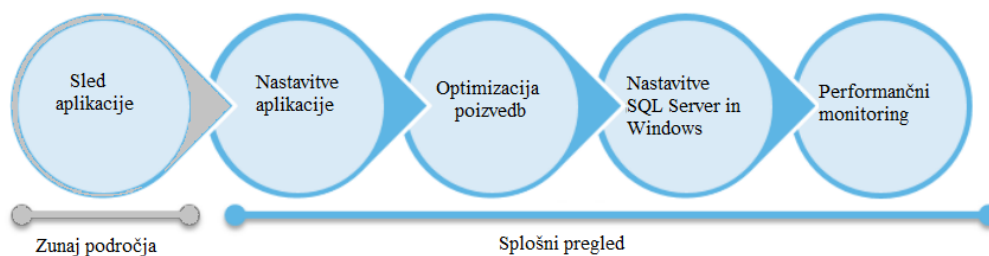


**Slika 6.5:** Ciljno usmerjena performančna optimizacija

### 6.2.3 Performančna analiza (Q22 )

Cilj performančne analize je identificirati glavne razloge performančnih težav in predlagati ustrezne izboljšave, ki bodo pripomogle k boljši učinkovitosti aplikacije.

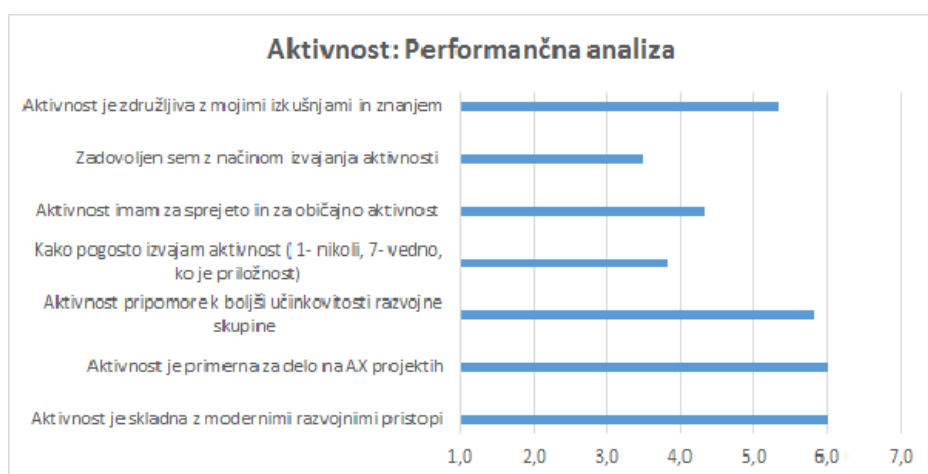
Pri performančni analizi se uporabljata dva različna pristopa. Ciljno usmerjena performančna optimizacija se osredotoča na optimizacijo specifičnega znanega problema. Splošna performančna optimizacija sistem obravnava kot celoto in predlaga možnosti za izboljšave. Glede na vrsto problema se izbere ustrezeni pristop pri performančni analizi. Področja performančne analize MS Dynamics AX lahko razdelimo na: nastavitve aplikacije, AOS Server nastavitve, nastavitve klienta, SQL Server nastavitve, nastavitve podatkovne baze in optimizacijo poizvedb.



**Slika 6.6:** Splošna performančna optimizacija



Aktivnost performančna analiza smo ovrednotili kot nesprejet EMR. Inženirji redko izvajajo performančno analizo. Tehnološka dimenzija EMR je pozitivno ovrednotena. Različna orodja, gradiva in smernice so s strani MS podrobno opisana in predpisana. Sociološka komponenta pa je ovrednotena podpovprečno, saj inženirji ne zaznavajo performančne analize za običajno aktivnost, zahteva specifična znanja in izkušnje in zato med člani razvojne skupine ni sprejeta.





## Poglavje 7

# Sklepne ugotovitve

Cilj diplomskega dela je ovrednotenje razvojnega procesa v opazovani razvojni skupini z uporabo modela za vrednotenje socio-tehničnih vidikov metodologije razvoja programske opreme. Iz celotne množice lastnosti, ki jih ponuja model, smo skladno z navodili modela izbrali tiste, ki so bile za vrednotenje podjetja najbolj relevantne, pri čemer smo se osredotočili zlasti na zadovoljstvo, sprejetost, združljivost, učinkovitost in skladnost.

Zastavljeni cilj diplomskega dela smo uspešno dosegli in organizaciji podali nekaj novih informacij o izvajajoči se metodologiji znotraj opazovane razvojne skupine. Zaznani trend naraščajoče potrebe po novih inženirjih in večanju števila zaposlenih na opazovanem oddelku, narekuje potrebo po novem in bolj specifičnem definiranju vlog inženirjev v razvojnih skupinah. Trenutna razporeditev dela in izvajanje EMR se nam ne zdi optimalno razporejena med člani razvojnih skupin. Način in ureditev dela ne predvidevata specializacije posameznika in zato posamezni inženirji izvajajo velik nabor zaznanih EMR, vključenih v naši raziskavi. Prav to dejstvo nam je otežilo pridobiti poglobljene informacije za vrednotenje tehnične ustreznosti EMR.

Dobljeni rezultati z merilnim instrumentom so bili skladni z našimi pričakovanji. Od skupaj zaznanih 24 EMR jih je večina razvrščenih na pozitivnem delu skale vrednotenja dimenzije. Umestitev posameznih EMR v kvadrante nam je podalo dobre smernice za nadaljnjo obravnavo, kako in katere EMR

najprej izboljšati. V opazovani organizaciji se je vodstvo odločilo, da bodo za člane razvojnih skupin uvedli interna izobraževanja, kjer bodo prve izbrane tematike šolanja povezane z zaznanimi neučinkovitimi EMR. Razpisali so termine in teme za naslednja področja: razvoj SSRS, razvoj OLAP, performančna analiza in vzpostavitev in vzdrževanje infrastrukture.

Predmet nadaljnjih raziskav bi bile izvedbe prilagoditev na ocenjenih neuporabnih, neučinkovitih ter nesprejetih EMR po osnovi načel in priporočil agilnih pristopov. Po prilagoditvi metodologije bi vrednotenje, ki je bil predmet te diplomske naloge, ponovno izvedli. Za ponovno vrednotenje in dopolnitve metodologije bi organizaciji predlagali postaviti skrbnika metodologije ter skrbnika vlog članov razvojnih skupin.

# Literatura

- [1] Avison, D. E. in Fitzgerald, G. Information systems development: methodologies, techniques and tools, 2003.
- [2] Bajec Marko, Krisper Marjan. Agilne metodologije razvoja informacijskih sistemov. Uporab. inform. (Ljubl.), apr. maj, jun. 2003, letn. 11, št. 2, str. 68-76, ilustr.
- [3] Centers for Medicare & Medicaid Services (CMS) Office of Information Service, Selecting a development approach, 2008.
- [4] Cockburg , Agile Software Development, 2002.
- [5] Eric J. Braude, Michael E. Bernstein, Software Engineering: Modern Approaches (2nd edition), 2010.
- [6] Maddison R.N., Information System Methodologies, 1983.
- [7] Vavpotič Damjan, Bajec Marko, An approach for concurrent evaluation of technical and social aspects of software development methodologies, 2009.
- [8] EMRIS - Enotna metodologija razvoja informacijskih sistemov. [Online]. Dosegljivo:  
<http://www2.gov.si/mju/emris.nsf>. [Dostopano 1. 4. 2016].
- [9] Manifesto for Agile Software Development. [Online]. Dosegljivo:  
<http://agilemanifesto.org/>. [Dostopano 1. 5. 2016].

- [10] Delivering large-scale IT projects on time, on budget, and on value [Online]. Dosegljivo:  
<http://www.mckinsey.com/business-functions/business-technology/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value>. [Dostopano 1. 5. 2016].
- [11] Scrum. [Online]. Dosegljivo:  
<http://www.scrumalliance.org/>. [Dostopano 1. 5. 2016].
- [12] Software Development Methodologies. [Online]. Dosegljivo:  
<http://www.itinfo.am/eng/software-development-methodologies/>. [Dostopano 10. 4. 2016].
- [13] Software development process. [Online]. Dosegljivo:  
[https://en.wikipedia.org/wiki/Software\\_development\\_process](https://en.wikipedia.org/wiki/Software_development_process). [Dostopano 1. 4. 2016].